FORM PTO-1390 (REV 10-94) U S DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371

ATTORNEY'S DOCKET NUMBER: 13267.2USWO

U S APPLICATION NO (If known, see 37 C F R 1.5): Unknown 09/600297

INTERNATIONAL APPLICATION NO.	INTERNATIONAL FILING DATE	PRIORITY DATE CLAIMED
PCT/SG98/00003	16 January 1998	

TITLE OF INVENTION

A METHOD OF DATA STORAGE AND APPARATUS THEREFOR

APPLICANT(S) FOR DO/EO/US

HU et al.

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. [X] This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. [ ] This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. [X] This express request to begin national examination procedures (35 U.S.C. 371(f)) at any time rather than delay examination until the expiration of the applicable time limit set in 35 U.S.C. 371(b) and PCT Articles 22 and 39(l).
4. [ ] A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date.
5. [X] A copy of the International Application as filed (35 U.S.C. 371(c)(2))
   a. [X] is transmitted herewith (required only if not transmitted by the International Bureau).
   b. [X] has been transmitted by the International Bureau.
   c. [ ] is not required, as the application was filed in the United States Receiving Office (RO/US)
6. [ ] A translation of the International Application into English (35 U.S.C. 371(c)(2)).
7. [X] Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3))
   a. [ ] are transmitted herewith (required only if not transmitted by the International Bureau).
   b. [ ] have been transmitted by the International Bureau.
   c. [ ] have not been made; however, the time limit for making such amendments has NOT expired.
   d. [X] have not been made and will not be made.
8. [ ] A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
9. [X] An oath or declaration of the inventor(s) (35 U.S.C. 371 (c)(4)).
10. [ ] A translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

Items 11. to 16. below concern document(s) or information included:

11. [X] An Information Disclosure Statement under 37 CFR 1.97 and 1.98.
12. [ ] An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
13. [X] A FIRST preliminary amendment.
    [ ] A SECOND of SUBSEQUENT preliminary amendment.
14. [ ] A substitute specification.
15. [ ] A change of power of attorney and/or address letter.
16. [X] Other items or information: Form 1449, 6 references

U.S. APPLICATION NO. (If known, see 37 C.F.R. 1.5)	INTERNATIONAL APPLICATION NO.	ATTORNEY'S DOCKET NUMBER
Unknown 09/600297	PCT/SG98/00003	13267.2USWO

17. [X] The following fees are submitted:

CALCULATIONS PTO USE ONLY

BASIC NATIONAL FEE (37 CFR 1.492(a) (1)-(5)):

Search Report has been prepared by the EPO or JPO **$840.00**

International preliminary examination fee paid to U.S. Patent and Trademark Office (37 CFR 1.492(a)(1)) **$670.00**

No international preliminary examination fee paid to USPTO (37 CFR 1.482) but international search fee paid to USPTO (37 CFR 1.445(a)(2)) **$760.00**

Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(3)) paid to USPTO **$970.00**

International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(2)-(4) **$96.00**

ENTER APPROPRIATE BASIC FEE AMOUNT =				**$ 970.00**	
Surcharge of **$130.00** for furnishing the oath or declaration later than [ ] 20 [ ] 30 months from the earliest claimed priority date (37 CFR 1.492(e)).				**$**	
CLAIMS	NUMBER FILED		NUMBER EXTRA	RATE	
Total claims	35	-20 =	15	X **$18.00**	**$ 270.00**
Independent claims	3	-3 =		X **$78.00**	**$**
MULTIPLE DEPENDENT CLAIM(S) (if applicable)				+ **$260.00**	**$**
TOTAL OF ABOVE CALCULATIONS =					**$1,240.00**
Reduction by 1/2 for filing by small entity, if applicable. Verified Small Entity Statement must also be filed (Note 37 CFR 1.9, 1.27, 1.28).					**$**
SUBTOTAL =					**$1,240.00**
Processing fee of **$130.00** for furnishing the English translation later than [ ] 20 [ ] 30 months from the earliest claimed priority date (37 CFR 1.492(f).				+	**$**
TOTAL NATIONAL FEE =					**$1,240.00**
Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31). $40.00 per property				+	**$**
TOTAL FEES ENCLOSED =					**$1,240.00**
				Amount to be: refunded	**$**
				charged	**$**

a. [X] Check(s) in the amount of $1,240.00 to cover the above fees is enclosed.

b. [ ] Please charge my Deposit Account No. _____ in the amount of $ _____ to cover the above fees. A duplicate copy of this sheet is enclosed.

c. [X] The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 13-2725.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:
Michael D. Schumann
MERCHANT & GOULD
P.O. Box 2903
Minneapolis, MN 55402-0903

SIGNATURE: James A. Larson

NAME: James A. Larson

REGISTRATION NUMBER: 40,443

S/N unknown

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Hu et al.	Docket No.:	13267.2USWO
Serial No.:	unknown	Filed:	concurrent herewith
Int'l Appln No.:	PCT/SG98/00003	Int'l Filing Date:	16 January 1998
Title:	A METHOD OF DATA STORAGE AND APPARATUS THEREFOR		

CERTIFICATE UNDER 37 CFR 1.10
'Express Mail' mailing label number: EL415942576US
Date of Deposit: 13 July 2000
I hereby certify that this correspondence is being deposited with the United States Postal Service 'Express Mail Post Office To Addressee' service under 37 CFR 1.10 on the date indicated above and is addressed to the Assistant Commissioner for Patents, Washington, D.C . 20231.

By: Linda McCormick
Name: Linda McCormick

PRELIMINARY AMENDMENT

Assistant Commissioner for Patents
Washington, D. C. 20231

Dear Sir:

In connection with the above-identified application filed herewith, please enter the following preliminary amendment:

ABSTRACT

Insert the attached Abstract page into the application as the last page thereof.

THE SPECIFICATION

A courtesy copy of the originally-filed PCT specification is enclosed herewith, but the World Intellectual Property Office (WIPO) copy should be relied upon if it is already in the U.S. Patent Office.

IN THE CLAIMS

Please delete the multiple dependencies from the following claims.

In claim 3, 1ine 1, please delete "claim 1 or claim 2" and insert --claim 1--.

In claim 4, line 1, please delete "any one of the preceding claims" and insert --claim 1--.

In claim 5, line 1, please delete "any one of the preceding claims" and insert --claim 1--.

In claim 6, line 1, please delete "any one of the preceding claims" and insert --claim 1--.

In claim 7, line 1, please delete "any one of the preceding claims" and insert --claim 1--.

In claim 8, line 1, please delete "any one of the preceding claims" and insert --claim 1--.

In claim 9, line 1, please delete "any one of the preceding claims" and insert --claim 1--.

In claim 11, line 1, please delete "any one of the preceding claims" and insert --claim 1--.

In claim 17, line 1, please delete "any one of the preceding claims" and insert --claim 1--.

In claim 20, line 1, please delete "any one of the preceding claims" and insert --claim 1--.

In claim 21, line 1, please delete "any one of the preceding claims" and insert --claim 1--.

In claim 24, line 1, please delete "any one of the preceding claims" and insert --claim 1--.

In claim 27, line 1, please delete "any one of the preceding claims" and insert --claim 1--.

In claim 30, line 1, please delete "any one of the preceding claims" and insert --claim 1--.

In claim 35, line 1, please delete "claim 33 or claim 34" and insert --claim 33--.

REMARKS

The above preliminary amendment is made to remove multiple dependencies from claims 3-9, 11, 17, 20, 21, 24, 27, 30 and 35.

Applicants respectfully request that the preliminary amendment described herein be entered into the record prior to calculation of the filing fee and prior to examination and consideration of the above-identified application.

If a telephone conference would be helpful in resolving any issues concerning this communication, please contact Applicants' primary attorney-of record, Michael D. Schumann (Reg. No. 30,422), at (612) 336.4638.

Respectfully submitted,

MERCHANT & GOULD P.C.
P.O. Box 2903
Minneapolis, Minnesota 55402-0903
(612) 332-5300

Dated: 13 July 2000

By _____
James A. Larson
Reg. No. 40,443

MDS(JAL)/klj

A METHOD OF DATA STORAGE AND APPARATUS THEREFOR

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to the field of data processing, and more particularly, but not exclusively, to a method and apparatus for implementing a system which is able to provide digital storage services for public or corporate users.

Description of the Related Art

The explosive growth in automatic information systems and their interconnections via "cyberspace" has increased the dependence of both organizations and individuals on the information processed, stored and communicated using these systems. Some of the information, hereinafter called digital valuables or valuables for short - whether it be text, graphics, animation, video, audio, all types of data, or software (such as source code or machine code) written in various programming languages, are very precious for a user - whether the user is an organisation or individual, and need be kept in a safe place from which the valuables can be retrieved reliably and securely at a later stage. Examples of such digital valuables are patients' medical records, financial transactions, various certificates, proprietary business data, electronic money, and legal documents.

Currently, a user's digital information is commonly stored in the user's computer or in a networked file server. There are several fundamental problems with such an approach in that, firstly, it lacks integrity protection. Digital information is inevitably in digital format. The information is easy to tamper with, and the modifications need not leave any trace on the physical medium. In many situations it is necessary for a user to verify that the information retrieved is indeed what was stored in the first place and any modification to the information by system faults or malicious process can be detected. The second problem is low reliability. A hard disk or floppy disk may crash, resulting in irreversible loss of data. A networked file server provides higher reliability than a local hard disk but loss of

data is not uncommon. Moreover, such servers lack adequate security protection and are not easily accessible to public users. The third problem with the current technology of storing information is that it has little confidentiality protection.

It is an object of the invention to alleviate at least one disadvantage of the prior art.

SUMMARY OF THE INVENTION

According to the invention in a first aspect, there is provided a digital data depository for storing digital data items for a user comprising:
data storage means;
a user account associated with the user;
means for communication with the user;
means for authentication of the user with the depository;
means for establishing a digital data transaction session in which the user is able to instruct storage or retrieval of a digital data item in association with the user's account;
means for encoding the data item into a plurality of parts, the parts being separately stored in the storage means;and
means for decoding the encoded data item.

According to the invention in a second aspect, there is provided a method of storing digital data items for a user comprising the steps of:
providing a user account associated with the user;
authenticating the identity of the user;
receiving a digital data item and an instruction from the user for the item to be stored in association with the user's account; and
encoding the data item into a plurality of parts and storing the parts separately.

According to the invention in a third aspect, there is provided a method of protecting digital data comprising:

providing a data depository in which digital data may be stored electronically;

providing for registration of users of the data depository, each user having an

account with the depository;

in response to a request from a user, opening a transaction session with the user in which the user and the depository authenticate each other and performing a transaction instructed by the user in respect of a digital data item, the transaction being selected by the user from a plurality of available transactions including storage of the item in or retrieval of the item from the depository.

The described embodiment of the present invention discloses a method for implementing a digital valuables depository system, for public or corporate users to store and to retrieve precious digital information. The described embodiment is the electronic analogy to the physical safe boxes provided by banks whereby customers can keep their precious belongings. There are two generic entities in the system, a Service Provider (SP) and a User, the SP operating and provides Digital Safe services to the User. To make use of the service, the User first registers with the SP to open an account. The User can then deposit digital valuables into, retrieve and delete them from its account, all being carried out in an authentic and secure manner.

The SP ensures high reliability in storing users' digital valuables. To store a valuable, the SP first encodes each valuable into N parts based on an encoding algorithm, and then stores the N parts into one or more data storage devices. To retrieve or copy a valuable which has been stored previously, the SP reads the N parts from corresponding storage devices, and recovers the valuable from the N parts based on a decoding algorithm. The encoding and decoding algorithms are chosen such that the original valuable can be recovered correctly even if some of the N parts are lost or corrupted. To avoid storage error/corruption accumulation, the system periodically checks the N parts of every stored valuables and recovers/corrects lost/corrupted parts when they are detected.

BRIEF DESCRIPTION OF THE DRAWINGS

An embodiment of the invention will now be described, by way of example, with reference to the accompanying drawings in which:

FIG. 1 is a schematic diagram of a data storage system, being an embodiment of the invention.

FIG. 2 shows a possible data structure of the User Account maintained by the Service Provider (SP) in the preferred embodiment of the present invention.

FIG. 3 shows the steps to allow the User accessing his account in the preferred embodiment of the present invention.

FIG. 4(a) illustrates a possible logical structure of the Transaction Request message of type Valuable Storage (VS_Req) in accordance with the preferred embodiment of the present invention.

FIG. 4(b) illustrates a possible logical structure of the Transaction Request message of type Valuable Copy (VC_Req) in accordance with the preferred embodiment of the present invention.

FIG. 4(c) illustrates a possible logical structure of the Transaction Request message of type Valuable Deletion (VD_Req) in accordance with the preferred embodiment of the present invention.

FIG. 4(d) illustrates a possible logical structure of the Transaction Request message of type Valuable Retrieval (VR_Req) in accordance with the preferred embodiment of the present invention.

FIG. 4(e) illustrates a possible logical structure of the Transaction Request message of type Account Status Report (ASR_Req) in accordance with the preferred embodiment of the present invention.

FIG. 4(f) illustrates a possible logical structure of the Transaction Request message of type Session Close (SC_Req) in accordance with the preferred embodiment of the present invention.

FIG. 5(a) illustrates a possible logical structure of the Transaction Response message of type Valuable Storage (VS_Resp) in accordance with the preferred embodiment of the present invention.

FIG. 5(b) illustrates a possible logical structure of the Transaction Response message of type Valuable Copy (VC_Resp) in accordance with preferred embodiment of the present invention.

FIG. 5(c) illustrates a possible logical structure of the Transaction Response message of type Valuable Deletion (VD_Resp) in accordance with the preferred embodiment of the present invention.

FIG. 5(d) illustrates a possible logical structure of the Transaction Response message of type Valuable Retrieval (VR_Resp) in accordance with the preferred embodiment of the present invention.

FIG. 5(e) illustrates a possible logical structure of the Transaction Response message of type Account Status Report (ASR_Resp) in accordance with the preferred embodiment of the present invention.

FIG. 5(f) illustrates a possible logical structure of the Transaction Response message of type Session Close (SC_Resp) in accordance with the preferred embodiment of the present invention.

FIG. 5(g) illustrates a possible logical structure of the Transaction Response message of type Error (ERR_Resp) in accordance with the preferred embodiment of the present invention.

FIG. 6 shows the flow diagram of a Transaction Response Program (TRP) used in the preferred embodiment of the present invention.

FIG. 7 illustrates the flow diagram of the first Valuable Dispersal Program (VDP) used in

the preferred embodiment of the present invention.

FIG. 8 shows the flow diagram of the first Valuable Recovery Program (VRP) used in the preferred embodiment of the present invention.

FIG. 9 illustrates the flow diagram of the second Valuable Dispersal Program (VDP) used in the preferred embodiment of the present invention.

FIG. 10 shows the flow diagram of the second Valuable Recovery Program (VRP) used in the preferred embodiment of the present invention.

FIG. 11 shows the flow diagram of the first Valuable Checking and Correction Program (VCCP) used in the preferred embodiment of the present invention.

FIG. 12 shows the flow diagram of the second Valuable Checking and Correction Program (VCCP) used in the preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Methods for implementing a digital data depository system,hereinafter referred to as Digital Safe, are described. The system is an electronic analogy to the physical safe boxes provided by banks whereby customers can keep their precious belongings. In the following description, numerous specific details are set forth such as logical structures of digital information and program steps, etc. in order to provide a thorough understanding of the present invention. It will be obvious to one skilled in the art that the present invention may be practised without these specific details. In other instances, well known steps as those involved in public key and symmetric key cryptosystem operations, in computing digest of a message using a one-way hash function, in digital signature generation and verification, in encoding of a data item into a codeword in an error-control coding scheme, in error-and-erasure-correction decoding, in erasure-correction decoding, are not shown in order not to obscure the present invention.

Notation and Nomenclature

The detailed description with respect to the implementation and operations of the Digital Safe system is presented partially in terms of algorithmic and symbolic representations of operations on data bits within the computer memory. These algorithmic descriptions and representations are the means used by those skilled in the art of data processing to most effectively convey the substance of their work to others skilled in the art.

An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. These steps are those require physical manipulation of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, and otherwise manipulated. In this case, the physical quantities are voltage or current signals which correspond to the digital valuables/information being processed. It proves convenient at times, principally for reason of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, items, fields, numbers or the like. It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities.

Furthermore, the manipulations performed are often referred to in terms such as adding or comparing, which are commonly associated with the mental operations performed by a human operator. No such capability of a human operator is necessary, or desirable. In most cases, in any of the operations described herein which form part of the present invention, the operations are machine operations. Useful machines for performing the operations of the present invention include general purpose digital computers or similar devices such as digital signal processors. In all cases, it should be borne in mind that there is a distinction between the method operation in operating a computer or other apparatus and the method of computation itself. The embodiment of the present invention to be described relates to method steps for secure and reliably storing users' digital valuables into and then subsequently retrieving them from a data depository system maintained and operated upon by a service provider.

The embodiment of the present invention also relates to an apparatus for performing these operations. This apparatus may be specially constructed for the required purpose or it may comprise general purpose computers as selectively activated or reconfigured by a computer program stored in the computers. The algorithms presented herein are not inherently related to any particular computer or other apparatus. In particular, various general purpose machines may be used with programs written in accordance with the teachings herein, or it may prove more convenient to construct specialized apparatus such as digital signal processor to perform the required method steps. The required structure for a variety of these machines would appear from the description given below.

GENERAL SYSTEM CONFIGURATION

A general model of the Digital Safe system is shown in FIG. 1. in which the Service Provider (SP) 10 provides data depository services to User 30 via a transmission channel 20. The SP system is responsible for storing and retrieving the User's digital valuables in a secure, reliable, and authenticated manner. A digital valuable, or valuable for short, has an unique identifier and is a self-contained entity. There can be various types of valuables including but not restricted in form to text, graphics, animation, video, audio, software, or any combination thereof. The transmission channel 20 represents the means and more specifically the media through which communication messages are exchanged between the SP 10 and the User 30. Such messages include the User's requests to the SP over paths 25 and 15 and the SP's responses to the User over paths 15 and 25. The transmission channel 20 includes but is not limited to any communication means or media such as computer networks, radio links, satellite links, diskettes or other storage medium. It should be understood by one skilled in the art that the term User is interchangeable with any user of information.

The described embodiment teaches methods for securely and reliably storing Users' valuables into and then retrieving them from the User's account maintained by the SP. These methods will be described with reference to specific steps of manipulating information. For one skilled in the art, it is obvious that some of these steps shall be best automated by, for example, implementing them as a special purpose software, which is

usually called a server, running on general purpose computers. It is clear that an information provider could simultaneously initiate multiple executions of the server to serve multiple end users. It is also clear that there may exist multiple SPs. For example, there may be one SP per organization or per district.

For clarity of presentation, the description below will elaborate on the model having one SP and one User. It is also clear that a User may also be another SP.

PREFERRED EMBODIMENT OF THE PRESENT INVENTION

1. Overall System Set-Up

Referring to FIG. 1, prior to using the Digital Safe services provided by the SP 10, the User 30 registers itself to the SP. During this registration process, the User authenticates itself to the SP by whatever means as required by the SP. The User agrees to the terms of a service contract. Such a contract contains at a minimum the identities, addresses of both the User and the SP, and the types of services to be provided by the SP. It may contain the public keys of the User and the SP, respectively. These keys are selected by the respective party based on certain public-key cryptosystems (PKCs). It may contain secret keys shared by the Users and the SP. The Digital Safe system relies on cryptographic protocols for mutual authentication between the User and the SP and for protecting the confidentiality and integrity of the User's valuables. Such cryptographic protocols require the User to possess cryptographic keys or secrets to operate. The contract may also specify the procedures for the User to recover lost cryptographic keys or secrets.

Message authentication code (MAC) is used to protect the integrity of a message. A MAC of a message can be generated using a symmetric key cryptosystem with the message and a secret value as inputs or using a one-way hash function with the message and a secret value as inputs. Since both digital signature and MAC are used for message integrity protection and authentication, we will refer them as Integrity Check (IC). For further references on PKC, symmetric key cryptosystem, generation and verification of digital

signature, one-way hash functions, generation and verification of MAC, and public key certificate, see D. E. R. Denning, Cryptography and Data Security, Addition-Wesley, Reading, MA, 1983; W. Stallings, Network and Internetworks Security - Principles and Practice, Prentice Hall, Englewood Cliffs, NJ, 1995; and C. Kaufman, R. Perlman and M. Speciner, Network Security - Private Communication in a Public World, PTR Prentice Hall, Englewood Cliffs, NJ, 1995.

At the end of the User registration, the SP sets up an account for the User. There are at least two types of accounts. In the first type, called flat-rate account, the User's account is allocated a fixed data storage quota and time interval (which may be extended or reduced at the User's request) and the service charge might be at a flat rate. In the second type, called flexible-rate account, the User's account has no fixed data storage quota and the User is charged by usage.

The User's account might be represented by a data structure shown FIG. 2, which includes a unique Account number (Acc_No) 50, the User's identity (U_ID) 52, contact information 54, optionally public keys or public key certificates 56, and information items 58 related to valuables stored in the account. Possible information items 58 are valuable identity (V_ID), type, size, time of submission, a protection flag (P_FLAG), an optional storage interval for flexible-rate account and pointers to locations where the N parts of the valuable are stored.

The P_FLAG takes two possible values, Encryption_Required and Encryption_Not_Required. If P_FLAG is set to Encryption_Required, then the valuable is encrypted by the SP and a pointer to the decryption key is kept in the account for each SP encrypted valuable. If P_FLAG is set to Encryption_Not_Required, then the valuable is not encrypted by the SP. The information items related to stored valuables will be empty initially and will be updated every time the User updates its account or when a valuable's storage interval expires. It should be noted that all the items in the User's account can be made visible to the User except the pointers to decryption keys and the pointers to storage locations, which are only accessible to and modifiable by the SP.

2. User Transaction Session with Service Provider

The Digital Safe services provided by the SP 10 to the User 30 include Valuable Storage (VS), Valuable Copy (VC), Valuable Retrieval (VR), Valuable Deletion (VD), and Account Status Report (ASR). The VS service lets the User store new valuables in its account; the VC service allows the User to get copies of its previously stored valuables from its account, the VD service lets the User remove one or more valuables from its account; the VR service permits the User to get copies of previously stored valuables and at the same time delete those valuables from his account; and the ASR service provides the User with its most recent account status report.

FIG. 3 shows the steps a user follows in accessing its account in the preferred embodiment of the present invention. In FIG. 3, the User and the SP starts a transaction session by running a mutual authentication and a session key exchange protocol. The User 10 prepares an Open Request (O_Req) message and sends the O_Req to the SP in step 100. The O_Req message is used to initiate the communications between the User and the SP, and more importantly, to authenticate the User to the SP. The O_Req contains at least the User's account number Acc_No 50 and/or identity U_ID 52. It is integrity protected by an IC (i. e., either the User's digital signature or a MAC) which can be verified by the SP. Upon receiving the O_Req, the SP verifies its validity in step 110. If it is not valid, an alert signal is sent to the User and the SP in step 120. If it is valid, the SP generates an Open Response (O_Resp) message, sends it to the User and opens a session with the User in step 130. The O_Resp is used to authenticate the SP to the User. This is integrity protected by an IC (i. e., either the SP's signature or a MAC) which can be verified by the User. After receiving O_Resp, the User checks its validity in step 140. An alert signal is sent to the User and/or the SP in step 150 if the Response is not valid; otherwise, the User proceeds to step 160. The objectives of O_Req and O_Resp are mutual authentication of the User and the SP, to negotiate a unique session identifier to bind the User's transaction with his account, and to negotiate session keys for encrypting and integrity protecting the follow on message exchanges during the entire session if the communication path between the User and the SP is not secure. There are many authentication and key distribution protocols in the literature (see, for example, C. Kaufman, R. Perlman, and M. Speciner, Network

Security - Private Communication in A Public World, PTR Prentice Hall, Englewood Cliffs, NJ. 1995) which fulfil these requirements. Such protocols can be used in place of O_Req and O_Resp.

Referring again to FIG. 3, following the successful opening of the transaction session, the User generates and sends a Transaction Request (T_Req) in step 160 to the SP. There are five types of T_Req messages corresponding to the five types of services: Valuable Storage Request (VS_Req), Valuable Copy Request (VC_Req), Valuable Deletion Request (VD_Req), Valuable Retrieval Request (VR_Req), Account Status Report Request (ASR_Req). In addition, there is a Session Close Request (SC_Req) which is used by the User to request closing of the current session.

The possible logical structures of the six T_Req types in accordance with the preferred embodiment of the present invention are given in FIG. 4(a) - 4(f).

FIG. 4(a) shows the structure of VS_Req. This logical structure comprises a session identifier (S_ID) 170 which was negotiated during steps 100 and 110 in FIG. 3 and is used here to identify uniquely the present session, a transaction type field VS_Req 175 which identifies the type of T_Req as Valuable Storage, a valuable identifier V_ID 180 which shows the identity of the submitted valuable, a field P_FLAG 182 which takes two possible values Encryption_Required and Encryption_Not_Required. The Encryption_Required value indicates that the User's valuable be encrypted by the SP before it is stored while the Encryption_Not_Required value indicates that the User's valuable be not encrypted by the SP before it is stored. The VS_Req message also contains a field V_By 185 which is the submitted valuable body, an optional field V_SI 188 which specifies the storage time interval of the submitted valuable (note that this field is not needed for flat-rate account), a freshness identifier F_ID_U 190 which guarantees that the message is fresh or in sequence, and a message IC field IC_U 195. The V_By 185 may be in plaintext or in ciphertext. The latter is preferred if the User wants to keep the valuable confidential only to itself. The V_By contains the User's digital signature which can be used by either the User, or the SP, or an independent judge to verify the authenticity of the valuable. To simplify the presentation, only one V_ID and one V_By are shown here. In general, the

message in FIG. 4(a) may contain multiple V_ID and multiple V_By fields.

FIG. 4(b) depicts the structure of VC_Req. It comprises the S_ID 170, a transaction type field VC_Req 205 which identifies the type of T_Req as Valuable Copy, a valuable identifier V_ID 210 which shows the identity of the valuable to be copied, a freshness identifier F_ID_U 215 and a message IC field IC_U 220.

Possible logical structures of VD_Req and VR_Req are given in FIG. 4(c) and FIG. 4(d), respectively. In FIG. 4(c), the field VD_Req 225 indicates the type of T_Req as Valuable Deletion and the field V_ID 230 shows the identity of the valuable to be deleted. In FIG. 4(d), the field VR_Req 245 indicates the type of the T_Req as Valuable Retrieval and the field V_ID 250 identifies the name of the valuable to be retrieved.

To simplify the presentation, only one V_ID field is shown in FIG. 4(b) - 4(d). In general, each message in FIG. 4(b) - 4(d) may contain multiple V_IDs.

FIG. 4(e) and FIG. 4(f) show the structures of ASR_Req and SC_Req, respectively. In FIG. 4(e), the field ASR_Req 265 indicates the type of T_Req as Account Status Report, and the field ASR_Option 270 specifies the format of the status report. Possible values of ASR_Option are FULL (in which case the SP will send the User a full version report, as specified in FIG. 2) and SHORT (in which case the SP will send the User a User selectable short version report). In FIG. 4(f), the SC_Req 285 indicates that the T_Req is of type Session Close.

In FIG. 4(a) to FIG. 4(f), the F_ID_U and the IC_U fields are optional. They should be present if the communication path between the User and the SP is not integrity protected. The F_ID_U may be a timestamp, a sequence number, or a nonce (a non-repeating random number generated by SP and sent to the User in advance). The IC_U is computed over the entire message; it may be the User's digital signature generated using its private key or it may be a MAC generated using a secret key which is shared between the User and the SP. The User's digital signature must be used for the IC_U field whenever non-repudiation of origin of the T_Req messages is a requirement.

Referring to FIG. 3 again, upon receiving the T_Req in step 490, the SP processes it and prepares a Transaction Response (T_Resp) message using a Transaction Response Program (TRP) in step 500. There are at least seven types of T_Resp messages: Valuable Storage Response (VS_Resp), Valuable Copy Response (VC_Resp), Valuable Deletion Response (VD_Resp), Valuable Retrieval Response (VR_Resp), Account Status Report Response (ASR_Resp), Session Close Response (SC_Resp), and Error Response (ERR_Resp). FIG. 5(a) - 5(g) show possible logical structures of the seven Transaction Response types in accordance with the preferred embodiment of the present invention.

FIG. 5(a) is the structure of VS_Resp. It comprises a session identifier (S_ID) 170 which was negotiated during steps 100 and 110 in FIG. 3 and is used here to uniquely identify the present session, a field VS_Resp 300 which indicates the type of T_Resp as Valuable Storage, V_ID 180 taken from FIG. 4(a), a field VS_ACK 305 which indicates the success or failure of the transaction processing, a freshness identifier F_ID_SP 310 which guarantees that the message is fresh or in sequence, and a message IC field IC_SP 315.

FIG. 5(b) is the structure of VC_Resp. It consists of S_ID 170, a field VC_Resp 320 which identifies the T_Resp as type Valuable Copy, V_ID 210 taken from FIG. 4(b), the copied valuable V_By 325 as specified by V_ID, a field VC_ACK 327 indicating the success or failure of the transaction processing, the fields F_ID_SP 330 and IC_SP 335.

FIG. 5(c) is the structure of VD_Resp. It consists of S_ID 170, a field VD_Resp 340 which identifies the T_Resp as type Valuable Deletion, V_ID 230 taken from FIG. 4(c), a field VD_ACK 342 indicating the success or failure of the transaction processing, the fields F_ID_SP 345 and IC_SP 350.

FIG. 5(d) is the structure of VR_Resp. It consists of S_ID 170, a field VR_Resp 355 which identifies the T_Resp as type Valuable Retrieval, V_ID 250 taken from FIG. 4(d), the retrieved valuable V_By 360 as specified by V_ID, a field VR_ACK 362 showing the success or failure of the transaction processing, the fields F_ID_SP 365 and IC_SP 370.

FIG. 5(e) is the structure of ARS_Resp. It comprises S_ID 170, a field ASR_Resp 375

which identifies the T_Resp as type Account Status Report, a field ASR_Report 380 with its format as specified by ASR_Option 270 in FIG. 4(e), a ASR_ACK field 382 indicating the success or failure of the transaction processing, the fields F_ID_SP 385 and IC_SP 390.

FIG. 5(f) is the structure of SC_Resp. It consists of S_ID 170, a field SC_Resp 395 which identifies the T_Resp as type Session Close, a field CS_ACK 398 indicating the success or failure of the transaction processing, the fields F_ID_SP 400 and IC_SP 405.

FIG. 5(g) is the structure of ERR_Resp. It consists of S_ID 170, a field ERR_Resp 410 which identifies the T_Resp as type ERROR, a field T_Req_Type 415, a field ERR_Status 420, the fields F_ID_SP 425 and IC_SP 430. T_Req_Type indicates the type of T_Req with which the ERROR message is associated with. Possible values of T_Req_Type are VS_Req, VC_Req, VD_Req, VR_Req, ASR_Req, and SC_Req. ERR_Status shows the type of errors which takes possible values such as "Request Not Verified" and "Request Not Permitted".

In FIG. 5(a) to FIG. 5(g), the F_ID_SP and the IC_SP fields are optional. They should be present if the communication path between the SP and the User is not integrity protected. The F_ID_SP may be a timestamp, a sequence number, or a nonce (i. e., a non-repeating random number generated by the User and sent to the SP before hand). The IC_SP is computed over the entire message; it may be the SP's digital signature generated using its private key or it may be a MAC generated using a secret key shared between the SP and the User. The former must be used if non-repudiation of origin of T_Resp messages are required.

Referring again to FIG. 3, the SP sends the T_Resp to the User in step 1000. The T_Resp sent in step 1000 by the SP is received at the User side in step 1020. The User first verifies if T_Resp is valid in 1030, including verifying the freshness of the freshness identifier F_ID_SP and the validity of the integrity check IC_SP. If the answer is "No", an alert signal is sent to the User and/or the SP in step 1040 and the User/SP will take actions accordingly. Assuming that the output of step 1030 is "Yes", the User then checks to see

if the received T_Resp is of type SC_Resp. If yes, the current session is closed; otherwise, it accepts and outputs the results of the T_Resp message in step 1055 and then goes back to step 160.

3. Operations of the Transaction Response Program

FIG. 6 shows the flow diagram of a Transaction Response Program (TRP) used in the preferred embodiment of the present invention. Referring to FIG. 6, the T_Req received in step 490 in FIG.3 is first checked for its validity in step 520, including checking the validities of the freshness identifier F_ID_U and the integrity check IC_U, as well as checking if the requested operation in T_Req is permitted. If the T_Req is not valid, a T_Resp of type ERR_Resp is formed in step 530, where the format of the ERR_Resp is as specified in FIG. 5(g). Assuming that T_Req passes the checking in step 520, it is next inspected in step 540 to see if it is of type SC_Req. If the answer is "Yes", the TRP closes the current session and prepares a T_Resp of type SC_Resp in step 550 according to the format of the T_Resp given in FIG. 5(f). Assuming that the outcome in step 540 is "No", the T_Req is checked in step 560 to see if it is of type VS_Req. If yes, the TRP first processes and stores the submitted valuable using a Valuable Dispersal Program (VDP), updates the User's account (i. e., records the valuable's V_ID, time of storage, storage duration, pointers to storage locations, the value of P_FLAG and pointer to the decryption key if P_FLAG = Encryption_Required), and then prepares a T_Resp of type VS_Resp in step 570 according to the format as specified in FIG. 5(a). If step 560 outputs "No", the T_Req is next inspected to see if it is of type VC_Rep in step 580. If yes, the TRP first recovers the requested valuable using a Valuable Recovery Program (VRP) and then prepares a T_Resp of type VC_Resp in 590, where the format of VC_Resp is according to FIG. 5(b). Assuming that step 580 yields "No", the T_Req is next checked in step 600 to see if it is of type VD_Req, if the answer is "Yes", the TRP first deletes the valuable specified in T_Req and then prepares a T_Resp of type VD_Resp in step 610 according to the format as given in FIG. 5(c). Assuming that the outcome of step 600 is "No", the T_Resp is next checked in step 620 for type VR_Req. If yes, the TRP first recovers the requested valuable using VRP, delete the valuable from the User's account, and prepares a T_Resp of type VR_Resp according to the format of FIG. 5(d), all being performed in

step 630. If the output of step 620 is "No", the T_Req must be of type ASR_Req. In this case, the TRP prepares a T_Req of type ASR_Req, with its format as given in FIG. 5(e). It should be noted that the T_Resp messages prepared in steps 530, 550, 570, 590, 610, 630, and 640 are sent to the User in step 1000 in FIG. 3. The executions of the steps 540, 560, 580, 600, 620, and 640 do not have to follow the order given above; they may be carried out in any order according to the preference if the system designer.

4. Operations of the First Valuable Dispersal Program and the First Valuable Recovery Program

FIG. 7 illustrates the flow diagram of the first Valuable Dispersal Program (VDP) used in the preferred embodiment of the present invention. The valuable to be processed by the VDP program has the valuable identifier V_ID 180 and valuable body V_By 185 as specified by the VS_Req message shown in FIG. 4(a). The valuable contains the User's digital signature which can be checked by the User and the SP to verify its authenticity. The basic function of VDP is to transfer a submitted valuable into N parts based on an (N, K) error-control code C with symbols over Galois field GF(2^m), where m is a positive integer. For further reference on encoding and decoding of a (N, K) error-control code, see R. E. Blahut, Theory and Practice of Error Control Codes, Addison-Wesley, Reading, MA, 1983. Also see S. Lin and D. J. Costello, Jr., Error Control Coding: Fundamentals and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1983. Referring to FIG. 7, the User's digital signature on the valuable is first checked in step 700. If it is valid, the VDP proceeds to step 705; otherwise it goes to step 530 in Fig. 6. In step 705, the VDP checks if the P_FLAG 182 field of the VS_Req message is set to Encryption_Required. If "No", the program proceeds to step 715; otherwise, the valuable is encrypted under a cryptographic key in step 710. In step 715, the valuable or its ciphertext from step 710 is divided into q K-tuples, $X_i = (x_{i1}\ x_{i2}\ \ldots\ x_{iK})$ in step 715, for i = 1 to q, where xij is a symbol over GF(2^m). Each K-tuple X_i, i = 1 to q, is then encoded into a codeword $Y_i = (y_{i1}\ y_{i2}\ \ldots\ y_{iN})$ of the (N, K) error-control code C in step 720. Next, the q codewords Y_i, for i = 1 to q, are rearranged into N q-tuples, $Z_j = (y_{1j}\ y_{2j}\ \ldots\ y_{qj})$ in step 725, for j = 1 to N. Finally the q-tuples Z_j, j = 1 to N, are stored into one or multiple data storage devices.

FIG. 8 shows the flow diagram of the first Valuable Recovery Program (VRP) which works in conjunction with the first VDP in the preferred embodiment of the present invention. To recover the valuable (with identifier V_ID) which was dispersed by the first VDP in FIG. 7, the VRP first reads the N q-tuples $Z'_j = (y'_{1j}\ y'_{2j} \ldots y'_{qj})$, for j = 1 to N, from the corresponding storage devices in step 740 using the storage location pointers in the User's account, where Z'_j is the read version of Z_j. If Z'_j can not be found due to various reasons such as storage device crash or corruption, Z'_j is regarded as a q-tuple of "erasures", i. e., all the symbols in $Z'_j = (y'_{1j}\ y'_{2j} \ldots y'_{qj})$ are marked as "erasure". Next, the VRP rearranges N q-tuples Z'_j, j = 1 to N, into q N-tuples $Y'_i = (y'_{i1}\ y'_{i2} \ldots y'_{iN})$, for i = 1 to q, in step 745. It should be noted that if Z'_j is a q-tuple of "erasures", the jth symbol y'_{ij} in Y'_i is an "erasure". The N-tuple $Y'_i = (y'_{i1}\ y'_{i2} \ldots y'_{iN})$ is an erroneous codeword of the (N, K) code C, i.e., a codeword corrupted by errors and erasures. The q erroneous codewords $Y'_i = (y'_{i1}\ y'_{i2} \ldots y'_{iN})$, i = 1 to q, are input one by one to an error-and-erasure-correction decoder based on the (N, K) code C, to produce q K-tuples $X'_i = (x'_{i1}\ x'_{i2} \ldots x'_{iK})$, i = 1 to q, in step 750. If the errors and the erasures in $Y'_i = (y'_{i1}\ y'_{i2} \ldots y'_{iN})$ are correctable by the decoder, then $X'_i = X_i$. Step 755 checks to see if the q N-tuples are decoded successful. If not, an alarm is raised in step 765; if yes, the outputs of step 750, X'_i, i = 1 to q, are concatenated in step 760. In step 770, the VRP reads the corresponding P_FLAG from the User's account and checks if it equals Encryption_Required. If "No", the program produces the required valuable (which is the output of step 760) in step 780; if "Yes", the program proceeds to step 775 where it decrypts the output of step 760 using the appropriate decryption key. The result of step 775 is the required valuable which is output in step 780.

Let d denote the minimum Hamming distance of the (N, K) code C. Further, let e denote the number of erasures (i. e., lost symbols) and c the number of error symbols in $Y'_i = (y'_{i1}\ y'_{i2} \ldots y'_{iN})$, respectively. By the theory of error-control coding, the decoder output will be correct, i. e., $X'_i = X_i$, as long as $e + 2c < d$. In particular, for the so called maximum-distance codes, such as the Reed-Solomon codes, their minimum Hamming distance $d = N - K + 1$. Therefore, given K, an N can almost be selected, such that the probability that a valuable cannot be recovered is smaller than a pre-determined threshold.

5. Operations of the Second Valuable Dispersal Program and the Second Valuable

Recovery Program

FIG. 9 illustrates the flow diagram of the second Valuable Dispersal Program (VDP) used in the preferred embodiment of the present invention. The valuable to be processed by the VDP program has the valuable identifier V_ID 180 and valuable body V_By 185 as specified by the VS_Req message shown in FIG. 4(a). The valuable contains the User's digital signature which can be checked by the User and the SP to verify its authenticity. The basic function of VDP is to transfer a submitted valuable into N parts based on an (N, K) error-control code C with symbols over Galois field GF(2^m), where m is a positive integer. Referring to FIG. 9, the User's digital signature on the valuable is first checked in step 800. If it is valid, the VDP proceeds to step 810; otherwise it goes to step 530 in Fig. 6. In step 810, the VDP checks if the P_FLAG 182 of the VS_Req message is set to Encryption_Required. If "No", the program proceeds to step 815; otherwise, the valuable is encrypted under an encryption key in step 812. In step 815, the valuable (if P_FLAG = Encryption_Not_Required) or the output of step 812 (if P_FLAG = Encryption_Required) is divided into q K-tuples, $X_i = (x_{i1}\ x_{i2}\ \ldots\ x_{iK})$, for i = 1 to q, where xij is a symbol over GF(2^m). Each K-tuple X_i, i = 1 to q, is then encoded into a codeword $Y_i = (y_{i1}\ y_{i2}\ \ldots\ y_{iN})$ of the (N, K) error-control code C in step 820. The q codewords Y_i, for i = 1 to q, are rearranged into N q-tuples, $Z_j = (y_{1j}\ y_{2j}\ \ldots\ y_{qj})$ in step 825, for j = 1 to N. Next, an integrity check (as discussed under "Overall System Setup") IC_j is computed over Z_j, j = 1 to N, in step 830. Finally the (Z_j, IC_j), j = 1 to N, are stored into one or multiple data storage devices in step 835 with each Z_j being stored together with its corresponding IC_j.

FIG. 10 shows the flow diagram of the second Valuable Recovery Program (VRP) which works in conjunction with the second VDP in the preferred embodiment of the present invention. To recover the valuable (with identity D_ID) which was dispersed by the VDP of FIG. 9, the VRP first reads the N q-tuples $Z'_j = (y'_{1j}\ y'_{2j}\ \ldots\ y'_{qj})$ and the associated integrity checks IC'_j, for j = 1 to N, from the corresponding storage devices in step 840, where Z'_j is the read version of Z_j and IC'_j is the read version of IC_j. If Z'_j can not be found due to various reasons such as storage device crash or corruption, Z'_j is regarded as a q-tuple of "erasures", i. e., all the symbols in $Z'_j = (y'_{1j}\ y'_{2j}\ \ldots\ y'_{qj})$ are marked as "erasure". If Z'_j is found, it will be integrity checked using IC'_j. That is, an integrity check

is computed over Z'_j and the result is compared with IC'_j. If the two are equal, Z'_j is considered error free; otherwise, Z'_j is regarded as a q-tuple of "erasures". In step 845, the VRP checks to see if the number of erased Z'_js is less than d, the minimum Hamming distance of the (N, K) code. If not, it raises an alarm in step 850; if yes, the program proceeds to step 855 where the N q-tuples Z'_j, j = 1 to N, are rearranged into q N-tuples $Y'_i = (y'_{i1}\ y'_{i2} \cdots y'_{iN})$, for i = 1 to q. It should be noted that if Z'_j is a q-tuple of "erasures", the jth symbol y'_{ij} in Y'_i is an "erasure". The N-tuple $Y'_i = (y'_{i1}\ y'_{i2} \cdots y'_{iN})$ is a codeword corrupted by erasures. The q N-tuples $Y'_i = (y'_{i1}\ y'_{i2} \cdots y'_{iN})$, i = 1 to q, are input one by one to an erasure-correction decoder based on the (N, K) code C, to produce q K-tuples $X'_i = (x'_{i1}\ x'_{i2} \cdots x'_{iK})$, i = 1 to q, in step 860. As long as the number of erasures in Y'_i is less than d, the outputs of the decoder in step 860 will be correct. In step 865, the X'_i, i = 1 to q, are concatenated together. In step 870, the VRP reads the corresponding P_FLAG from the User's account and checks if it equals Encryption_Required. If "No", the program produces the required valuable (which is the output of step 865) in step 880; if "Yes", the program proceeds to step 875 where it decrypts the output of step 865 using the appropriate decryption key. The result of step 875 is the required valuable which is output in step 880.

6. Operations of Data Checking and Correction Programs

To avoid error accumulations in the N parts of the stored valuable, all stored data is checked periodically (in regular or irregular interests) by a Data Checking and Correction Program (DCCP). The DCCP checks if all the N parts of a valuable are intact. If not, it corrects the errors and recovers the missing parts.

There are two versions of DCCP. The first DCCP works in conjunction with the first VDP in section 4. Its flow diagram is shown in FIG. 11. In step 900, the DCCP collects the N q-tuples $Z'_j = (y'_{1j}\ y'_{2j} \cdots y'_{qj})$, j = 1 to N, associated with the valuable identified by a given D_ID. If Z'_j can not be found, DCCP marks Z'_j as a q-tuple of "erasures", j = 1 to N. It then rearranges Z'_j, j = 1 to N, into q N-tuples $Y'_i = (y'_{i1}\ y'_{i2} \cdots y'_{iN})$, i = 1 to q. In step 905, Y'_i is decoded into $X'_i = (x'_{i1}\ x'_{i2} \cdots x'_{iK})$, i = 1 to q, using an error-and-erasure-correction decoder of the (N, K) code C. Step 910 checks if all the q decoding operations in step 905 are successful. If not, an alarm is raised in step 915 and the program goes to

step 925; if yes, the DCCP proceeds to step 920. In step 920, X'_i is encoded into $Y''_i = (y''_{i1}\ y''_{i2} \cdots y''_{iN})$, i = 1 to q; the q N-tuples Y''_i, i = 1 to q, are rearranged into N q-tuples $Z''_j = (y''_{1j}\ y''_{2j} \cdots y''_{qj})$, j = 1 to N; finally, the old q-tuples $Z'_j = (y'_{1j}\ y'_{2j} \cdots y'_{qj})$ in the storage devices are replaced by the new ones $Z''_j = (y''_{1j}\ y''_{2j} \cdots y''_{qj})$, j = 1 to N. In step 925, the DCCP checks to see if there are more stored valuables need to be checked. If not, the program terminates; if yes, it gets the D_ID of the valuable to be checked in step 930 and goes back to step 900.

The second DCCP works in conjunction with the second VDP in section 5 and its flow diagram is shown in FIG. 12. For a given D_ID of a stored valuable, the DCCP collects the data segments (Z'_j, IC'_j), j = 1 to N, which are associated with the valuable in step 940. In step 945, the DCCP checks if all the N q-tuples $Z'_j = (y'_{1j}\ y'_{2j} \cdots y'_{qj})$ are found and if they all pass the integrity check based on IC'_j, j = 1 to N. If yes, the DCCP goes to step 975; if not, the DCCP proceeds to step 950 where the Z'_j is marked as a q-tuple of "erasures" if it is not found or if it fails to pass the integrity check. Step 955 checks to see if the number of erased Z'_js is less d, the minimum Hamming distance of the (N, K) code C. If not, the DCCP raises an alarm in step 960 and then goes to step 975; if yes, the program proceeds to step 965. In step 965, the Z'_j, j = 1 to N, are rearranged into $Y'_i = (y'_{i1}\ y'_{i2} \cdots y'_{iN})$, i = 1 to q, which are then decoded one by one using an erasure-correction decoder of the (N, K) code C, into $X'_i = (x'_{i1}\ x'_{i2} \cdots x'_{iK})$, i = 1 to q. In step 970, X'_i is encoded into codeword $Y''_i = (y''_{i1}\ y''_{i2} \cdots y''_{iN})$ of the (N, K) code C, i = 1 to q. The q codewords are then rearranged into q-tuples $Z''_j = (y''_{1j}\ y''_{2j} \cdots y''_{qj})$, j = 1 to N. Finally, the DCCP computes the integrity check IC''_j of Z''_j, and replace the old (Z'_j, IC'_j) with the new (Z''_j, IC''_j) in the storage devices, j = 1 to N. In step 975, the DCCP checks if there are any more valuables need to be checked. If not, the program terminates and if yes, it fetches D_ID of the valuable to be checked in step 980 and then goes back to step 945.

7. Procedures for Key Recovery

The User of the Digital Safe system needs to keep two sets of secret values, one for authentication to the SP in order to access its account and the other one for encrypting its valuables before submitting them to the SP. The former set of secret values are referred to

as User Authentication Secret Values (UASVs) and the latter set of secret values are referred to as Valuable Protection Secret Values (VPSVs). Examples of such secret values are private keys in public key cryptosystems and secret keys in symmetric key cryptosystems. All the secret values must be kept by the User in a secret and secure manner. An example of keeping a VPSV is to encrypt a valuable under the VPSV, encrypt the VPSV under a master key, and attach the encrypted VPSV with the valuable.

Loss of the UASVs prevents the User from accessing its account on-line. In the event of the User losing its UASVs, the User approaches pre-determined Key Escrow Agents to recover UASVs if UASVs are escrowed by such agents. Alternatively, the User will have to authenticate itself to the SP by other means (such as using paper documents) and upon authentication, the User and the SP separately or jointly establishing a new set of UASVs to replace the lost ones. The User from that point onwards uses the new UASVs to authenticate itself in order to access its account.

Loss of VPSVs prevents the User from decrypting its encrypted data or verifying the authenticity of the retrieved/copied valuables from its account. To prevent this from happening, the User should keep multiple copies of the VPSVs or make use of the key recovery services provided by a key escrow system. In the event of the User losing its VPSVs, the User approaches Key Escrow Agents to recover lost VPSVs. If the lost VPSVs are not escrowed, then valuables encrypted by the lost VPSVs will be lost too.

Claims

1. A digital data depository for storing digital data items for a user comprising:
data storage means;
a user account associated with the user; and
means for establishing a digital data transaction session in which the user is able to instruct storage or retrieval of a digital data item in association with the user's account;
means for encoding the data item into a plurality of parts, the parts being separately stored in the storage means;and
means for decoding the encoded data item.

2. A depository as claimed in Claim 1 wherein the data storage means comprises at least one data storage device, the parts being separately stored on the data storage device or devices.

3. A depository as claimed in Claim 1 or Claim 2 further comprising means for communication with the user.

4. A depository as claimed in any one of the preceding claims further comprising means for authentication of the user with the depository.

5. A depository as claimed in any one of the preceding claims further comprising means for authentication of the depository by the user.

6. A depository as claimed in any one of the preceding Claims wherein the user is able to instruct retrieval of a copy of the item in said transaction session.

7. A depository as claimed in any one of the preceding Claims wherein the user is able to instruct deletion of the digital data item in said transaction session.

8. A depository as claimed in any one of the preceding Claims wherein the user is able to instruct an account status report in said transaction session.

9. A depository as claimed in any one of the preceding Claims wherein the user's account has a data structure identifying the user and containing information identifying the data items stored therein.

10. A depository as claimed in Claim 9 wherein the information of each data item includes at least one of the type, size, time/date of submission, period of storage and pointers to the locations of the stored parts of the data item.

11. A depository as claimed in any one of the preceding Claims wherein the means for encoding:

a) divides the data item into a multiple of q K-tuples, denoted as $X_i = (x_{i1}\ x_{i2} \ldots x_{iK})$, i = 1 to q, where x_{ij} is a symbol over $GF(2^m)$ with m being a positive integer;

b) for i = 1 to q, encodes X_i into a codeword $Y_i = (y_{i1}\ y_{i2} \ldots y_{iN})$ using an (N, K) error-control code C, where Y_{ij} is a symbol over $GF(2^m)$;

c) rearranges Y_i, for i = 1 to q, into q-tuples $Z_j = (y_{1j}\ y_{2j} \ldots y_{qj})$, for j = 1 to N; and

d) stores the Z_j, for j = 1 to N, as said parts.

12. A depository as claimed in claim 11 wherein the means for decoding :

a) on inputting a data item identity, for j = 1 to N, reads $Z'_j = (y'_{1j}\ y'_{2j} \ldots y'_{qj})$ from the locations where Z_j was stored, where Z_j, j = 1 to N, are the parts of the data item as identified

b) rearranges Z'_j, for j = 1 to N, into N-tuples $Y'_i = (y'_{i1}\ y'_{i2} \ldots y'_{iN})$, for i = 1 to q;

c) decodes Y'_i using an error-and-erasure-correction decoder of the (N, K) code C to obtain $X'_i = (x'_{i1}\ x'_{i2} \ldots x'_{iK})$, for i = 1 to q;and

d) concatenates X'_i, for i = 1 to q to form the data item.

13. A depository as claimed in Claim 12 wherein the means for decoding:

e) at step (a), if Z_j cannot be found, assigns Z'_j as a q-tuple of erasures,such that in $Z'_j = (y'_{1j}\ y'_{2j} \ldots y'_{qj})$ each symbol is marked as an erasure; otherwise leaving Z'_j unchanged;

f) checks to see if all the decoding operations are successful and if not, raises an alarm.

14. A depository as claimed in Claim 11 wherein the means for encoding computes an integrity check IC_j over Z_j for j=1 to N and stores (Z_j, IC_j), for j=1 to N, as said parts.

15. A depository as claimed in Claim 14 wherein the means for decoding:
a) on inputting a data item identity, for j = 1 to N, reads $Z'_j = (Y'_{1j}\ Y'_{2j} \ldots Y'_{qj})$and IC'_j from the locations where (Z_j, Cj) was stored, where Z_j, j = 1 to N, are the parts of the data item as identified and C_j are the parts of the corresponding integrity check
b) rearranged Z'_j, for j = 1 to N, into N-tuples $Y'_i = (y'_{i1}\ y'_{i2} \ldots y'_{iN})$, for i = 1 to q;
c) decodes Y'_i using an error-and-erasure-correction decoder of the (N, K) code C to obtain $X'_i = (x'_{i1}\ x'_{i2} \ldots x'_{iK})$, for i = 1 to q;and
d) concatenates X'_i, for i = 1 to q to form the data item.

16. A depository as claimed in Claim 15 wherein the means for decoding:
e) at step (a), if Z_j cannot be found, assigns Z'_j as a q-tuple of erasures, such that in $Z'_j = (y'_{1j}\ y'_{2j} \ldots y'_{qj})$ each symbol is marked as an erasure; otherwise verifying the integrity of Z'_j based on IC'_j, if Z'_j fails the integrity verification, marking it as a q-tuple of erasures; otherwise leaving Z'_j unchanged;
f) checks to see if all the decoding operations are successful and if not, raises an alarm.

17. A depository as claimed in any one of the preceding claims further comprising means for encryption of the data item.

18. A depository as claimed Claim 17 wherein the user is able to instruct encryption, prior to encoding, of the data item to be stored during the transaction session.

19. A depository as claimed Claim 18 as dependent directly or indirectly on Claim 9 wherein the information of each data item includes an indication of whether or not the item is encrypted and a pointer to a decryption key.

20. A depository as claimed in any one of the preceding Claims further comprising means for decryption of an encrypted data item.

21. A depository as claimed in any one of the preceding Claims further comprising means for checking the encoded data items.

22. A depository as claimed in Claim 21 wherein the means for checking decodes, checks and reencodes the data item at intervals.

23. A depository as claimed in Claim 22 wherein the intervals are of fixed or variable period.

24. A depository as claimed in any one of the preceding Claims further comprising means for verifying the integrity of the data item and the data item includes an integrity check to be verified.

25. A depository as claimed in Claim 24 wherein the integrity check comprises a digital signature.

26. A depository as claimed in Claim 24 wherein the integrity check comprises a message authentication code.

27. A depository as claimed in any one of the preceding Claims wherein communication with the user during the transaction session is by means of a plurality of messages each associated with a transaction to be performed.

28. A depository as claimed in Claim 27 wherein at least one of said messages contains a freshness identifier.

29. A depository as claimed in Claim 28 wherein the freshness identifier comprises a timestamp, sequence number or a nonce.

30. A method of operating a depository as claimed in any one of the preceding Claims.

31. A method of storing digital data items for a user comprising the steps of:
providing a user account associated with the user;
authenticating the identity of the user;
receiving a digital data item and an instruction from the user for the item to be stored in association with the user's account;and
encoding the data item into a plurality of parts and storing the parts separately.

32. A method as claimed in Claim 31 further comprising the steps of:
receiving an instruction to retrieve a stored and encoded data item, decoding the data item and sending the data item to the user.

33. A method of protecting digital data comprising:

providing a data depository in which digital data may be stored electronically;

providing for registration of users of the data depository, each user having an account with the depository;

in response to a request from a user, opening a transaction session with the user in which the user and the depository authenticate each other and performing a transaction instructed by the user in respect of a digital data item, the transaction being selected by the user from a plurality of available transactions including storage of the item in or retrieval of the item from the depository.

34. A method as claimed in Claim 33 in which storage of the item includes encoding the item into a plurality of parts and storing the parts separately in the depository.

35. A method as claimed in claim 33 or Claim 34 further comprising the step of checking, at intervals, the integrity of data items stored in the depository.

ABSTRACT

A method and apparatus for implementing a digital valuables depository system, for public or corporation users to store and to retrieve valuable digital information electronically is described. The apparatus is the electronic analogy to the physical safe boxes provided by banks whereby customers can keep their valuable belongings. The digital valuable depository service is provided by a Service Provider (SP). To make use of the services, a user first opens a Digital Safe account with the SP. The user can then deposit digital valuables into, as well as retrieve, copy and delete them from its account, all being carried out in an authentic and secure manner. To store a digital valuable, the SP first encodes it into N parts based on an encoding algorithm, and then stores the N parts into one or more data storage devices. To retrieve or copy a digital valuable which has been stored previously, the SP reads the N parts based on a decoding algorithm. The encoding and decoding algorithms are selected such that the original digital valuable is recovered correctly even if some of the N parts are lost or corrupted. To avoid storage error/corruption accumulation, the system periodically checks the N parts of every stored digital valuables and recovers/corrects lost/corrupted parts when they are detected.

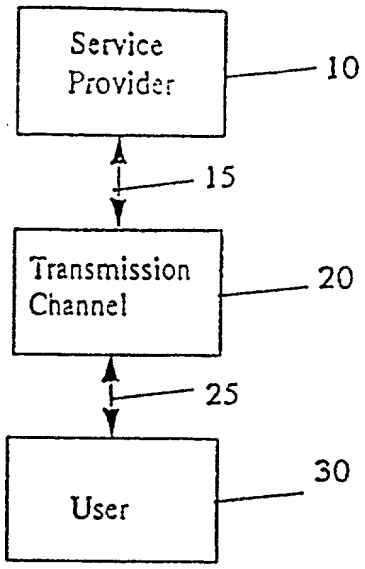
Service Provider
10
15
Transmission Channel
20
25
User
30

FIG. 1

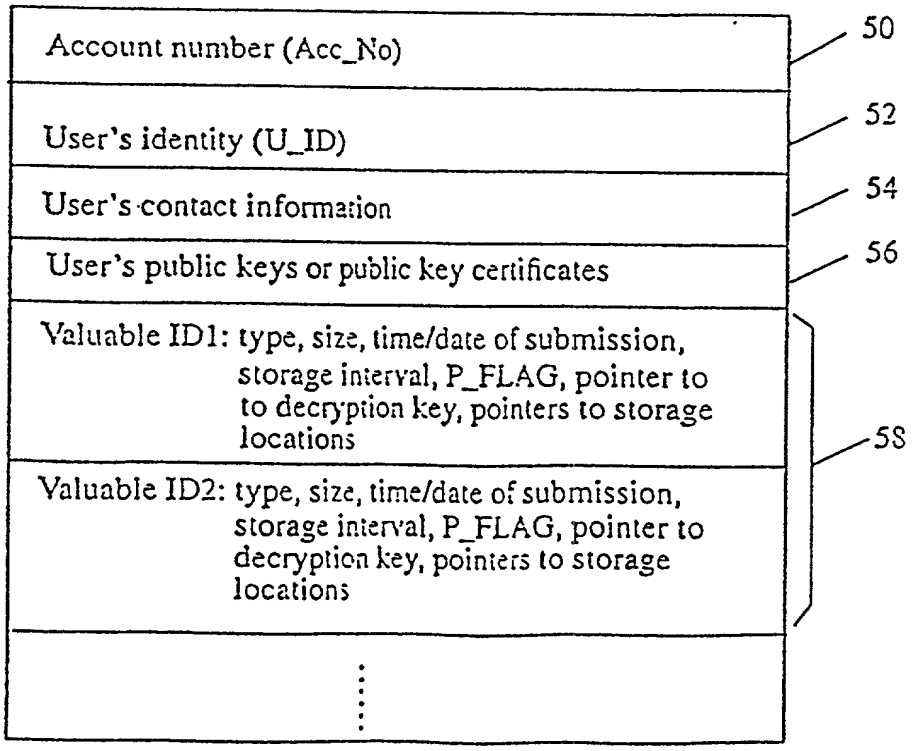
Account number (Acc_No)
50
User's identity (U_ID)
52
User's contact information
54
User's public keys or public key certificates
56
Valuable ID1: type, size, time/date of submission, storage interval, P_FLAG, pointer to to decryption key, pointers to storage locations
Valuable ID2: type, size, time/date of submission, storage interval, P_FLAG, pointer to decryption key, pointers to storage locations
58

FIG. 2

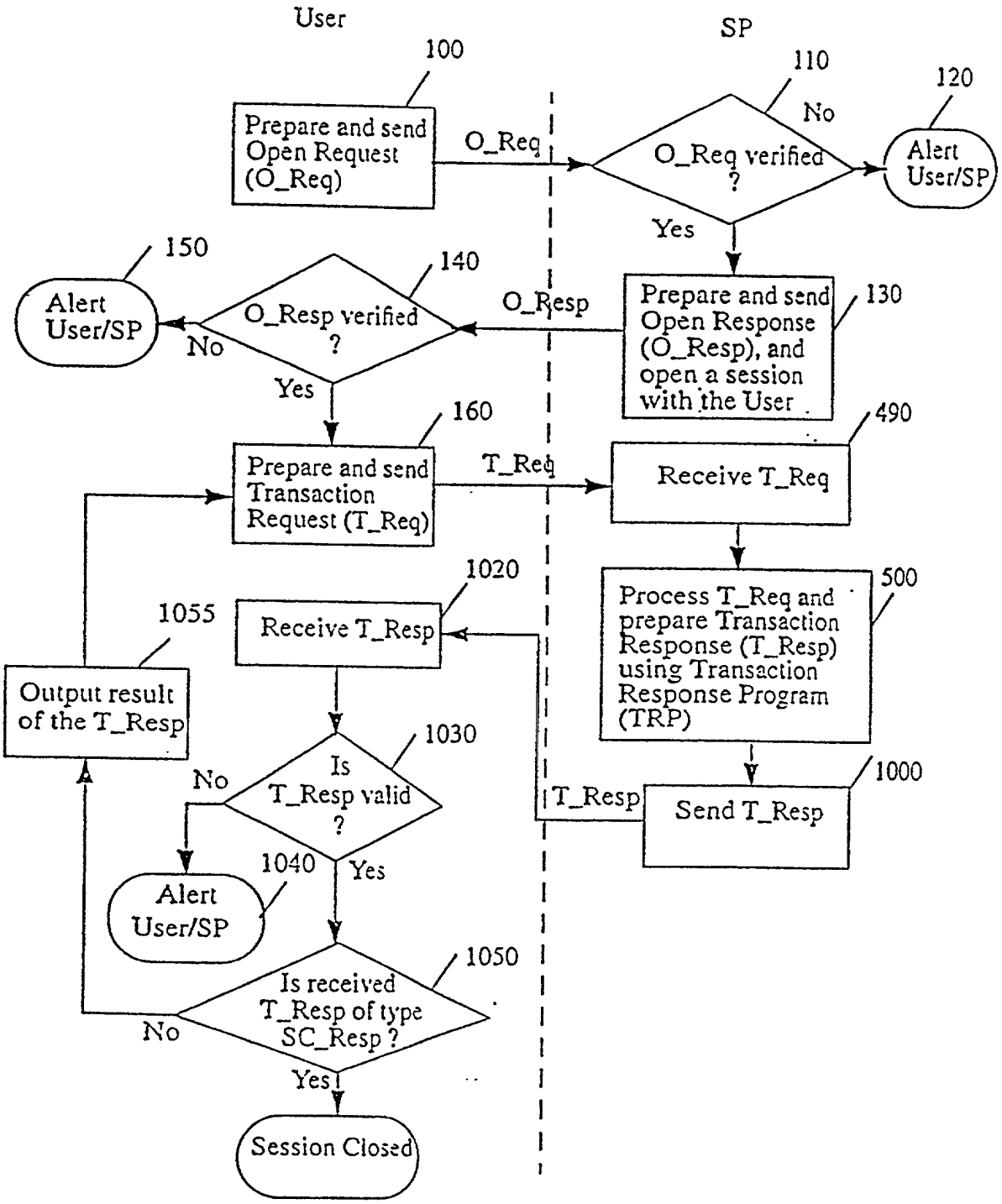
User
SP
100
Prepare and send Open Request (O_Req)
O_Req
110
O_Req verified ?
No
120
Alert User/SP
Yes
150
Alert User/SP
140
O_Resp verified ?
No
O_Resp
130
Prepare and send Open Response (O_Resp), and open a session with the User
Yes
160
Prepare and send Transaction Request (T_Req)
T_Req
490
Receive T_Req
1020
Receive T_Resp
500
Process T_Req and prepare Transaction Response (T_Resp) using Transaction Response Program (TRP)
1055
Output result of the T_Resp
1030
Is T_Resp valid ?
No
T_Resp
1000
Send T_Resp
Yes
1040
Alert User/SP
1050
Is received T_Resp of type SC_Resp ?
No
Yes
Session Closed

FIG. 3

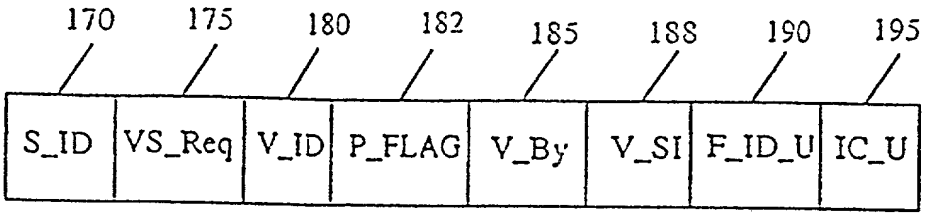

FIG. 4(a)

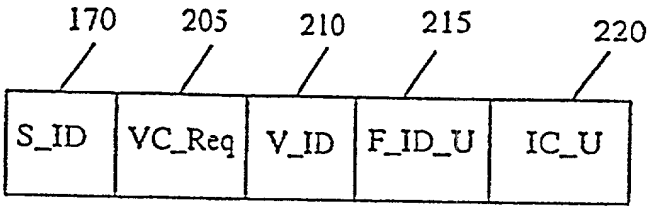

FIG. 4(b)

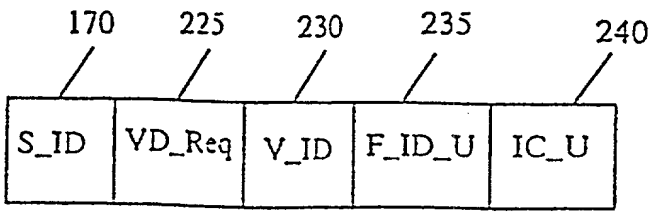

FIG. 4(c)

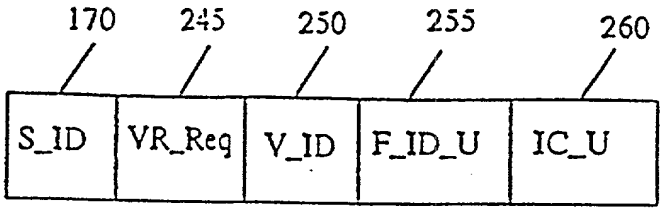

FIG. 4(d)

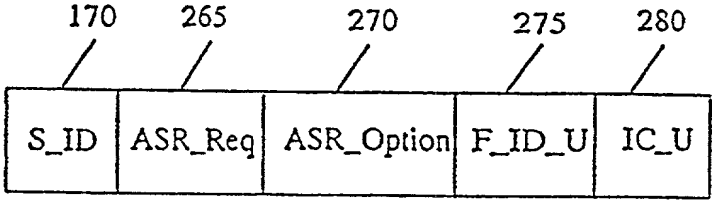

FIG. 4(e)

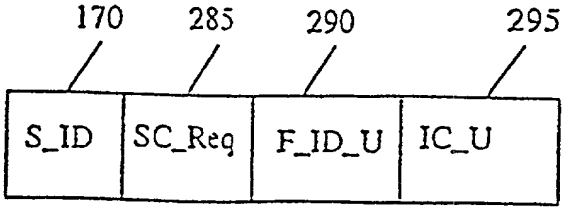

FIG. 4(f)

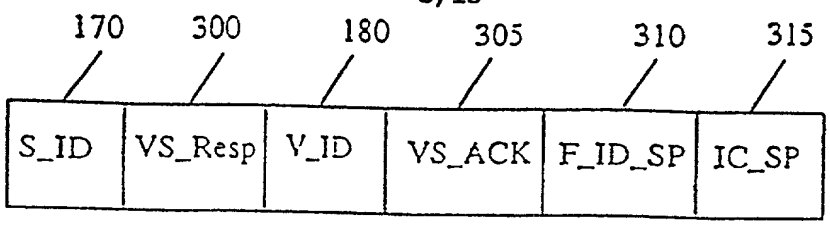

FIG. 5(a)

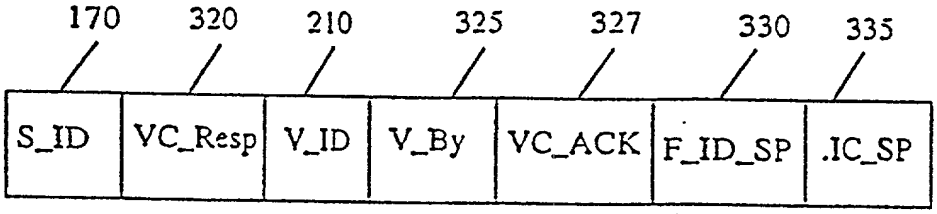

FIG. 5(b)

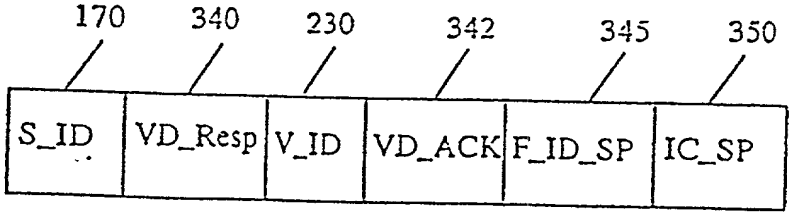

FIG. 5(c)

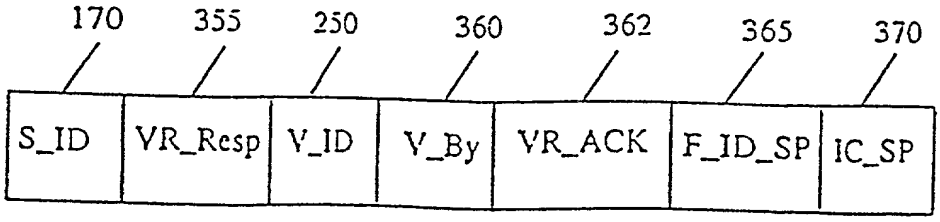

FIG. 5(d)

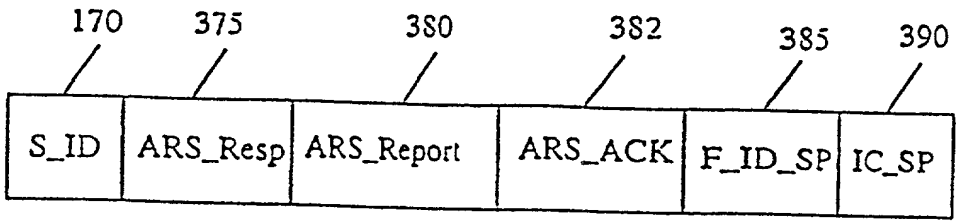

FIG. 5(e)

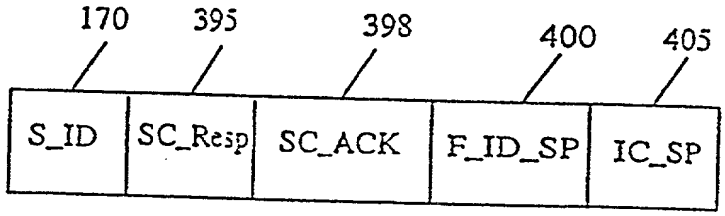

FIG. 5(f)

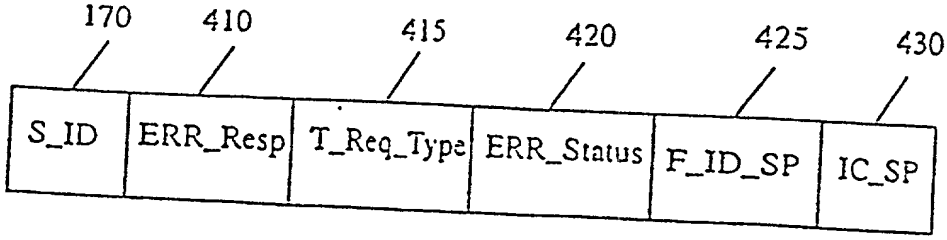

FIG. 5(g)

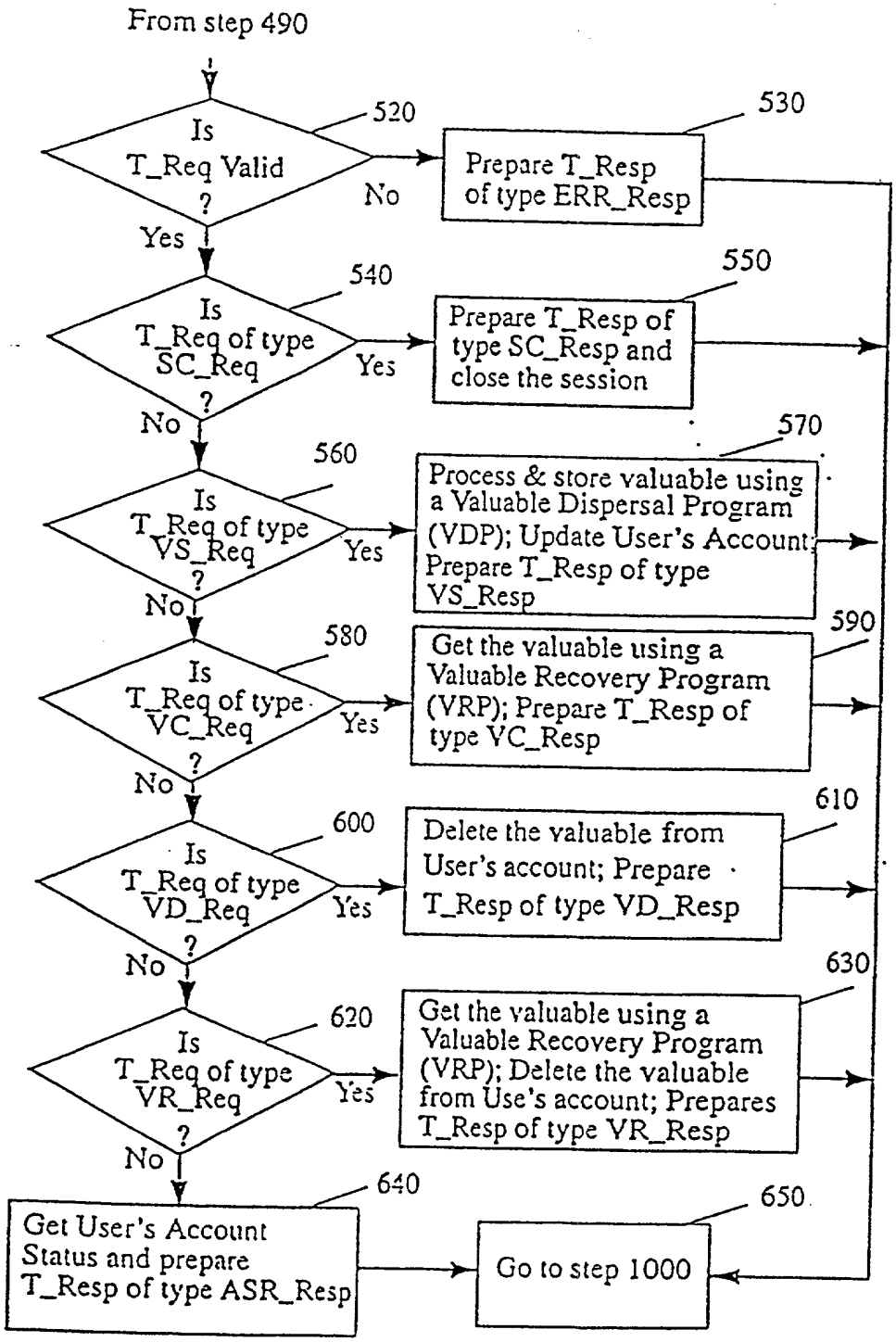
From step 490
Is T_Req Valid ?
520
No
Prepare T_Resp of type ERR_Resp
530
Yes
Is T_Req of type SC_Req ?
540
Yes
Prepare T_Resp of type SC_Resp and close the session
550
No
Is T_Req of type VS_Req ?
560
Yes
Process & store valuable using a Valuable Dispersal Program (VDP); Update User's Account; Prepare T_Resp of type VS_Resp
570
No
Is T_Req of type VC_Req ?
580
Yes
Get the valuable using a Valuable Recovery Program (VRP); Prepare T_Resp of type VC_Resp
590
No
Is T_Req of type VD_Req ?
600
Yes
Delete the valuable from User's account; Prepare T_Resp of type VD_Resp
610
No
Is T_Req of type VR_Req ?
620
Yes
Get the valuable using a Valuable Recovery Program (VRP); Delete the valuable from Use's account; Prepares T_Resp of type VR_Resp
630
No
Get User's Account Status and prepare T_Resp of type ASR_Resp
640
Go to step 1000
650

FIG. 6

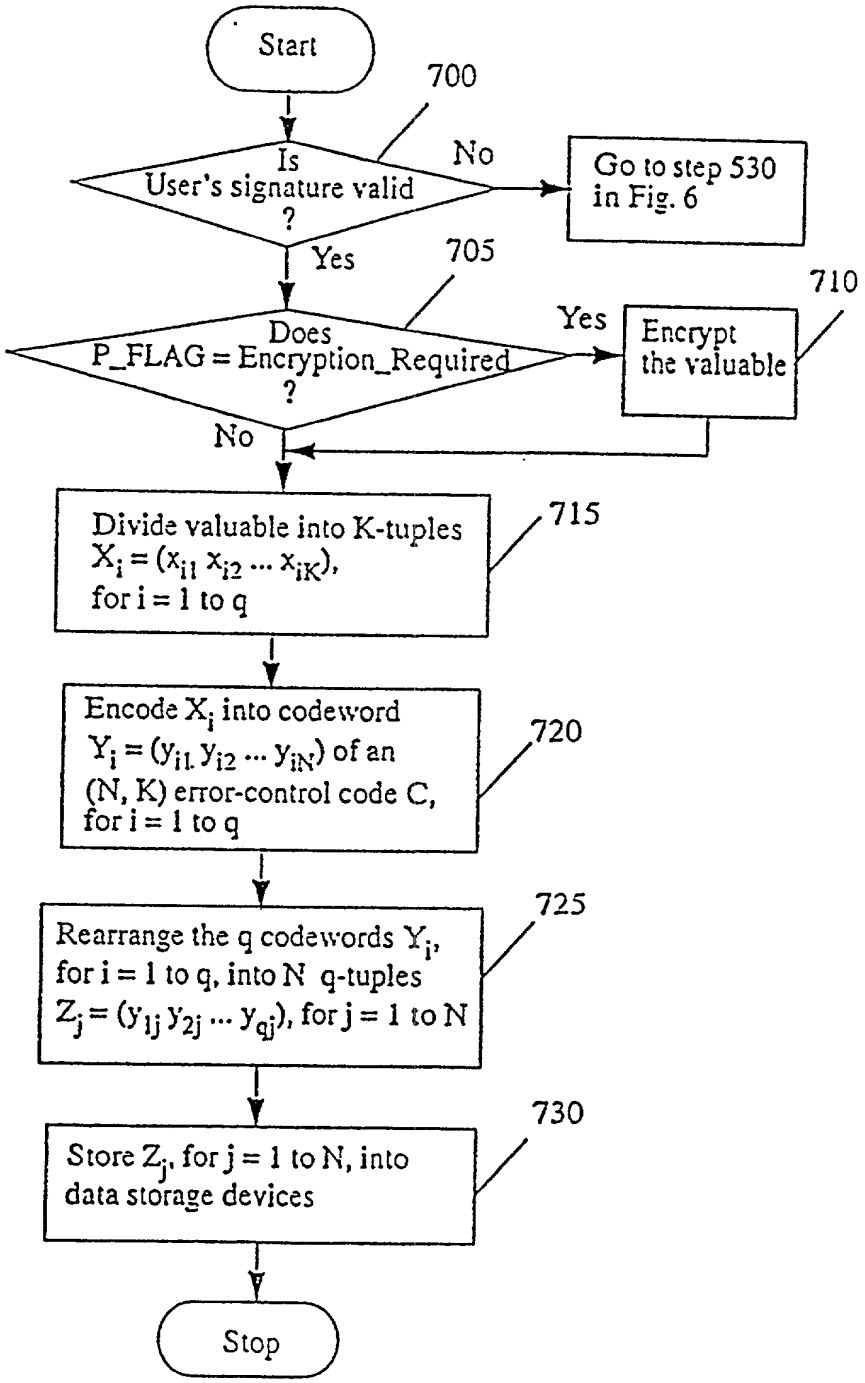
Start
700
Is User's signature valid ?
No
Go to step 530 in Fig. 6
Yes
705
Does P_FLAG = Encryption_Required ?
Yes
710
Encrypt the valuable
No
Divide valuable into K-tuples $X_i = (x_{i1}\ x_{i2} \cdots x_{iK})$, for i = 1 to q
715
Encode X_i into codeword $Y_i = (y_{i1}\ y_{i2} \cdots y_{iN})$ of an (N, K) error-control code C, for i = 1 to q
720
Rearrange the q codewords Y_i, for i = 1 to q, into N q-tuples $Z_j = (y_{1j}\ y_{2j} \cdots y_{qj})$, for j = 1 to N
725
Store Z_j, for j = 1 to N, into data storage devices
730
Stop

FIG. 7

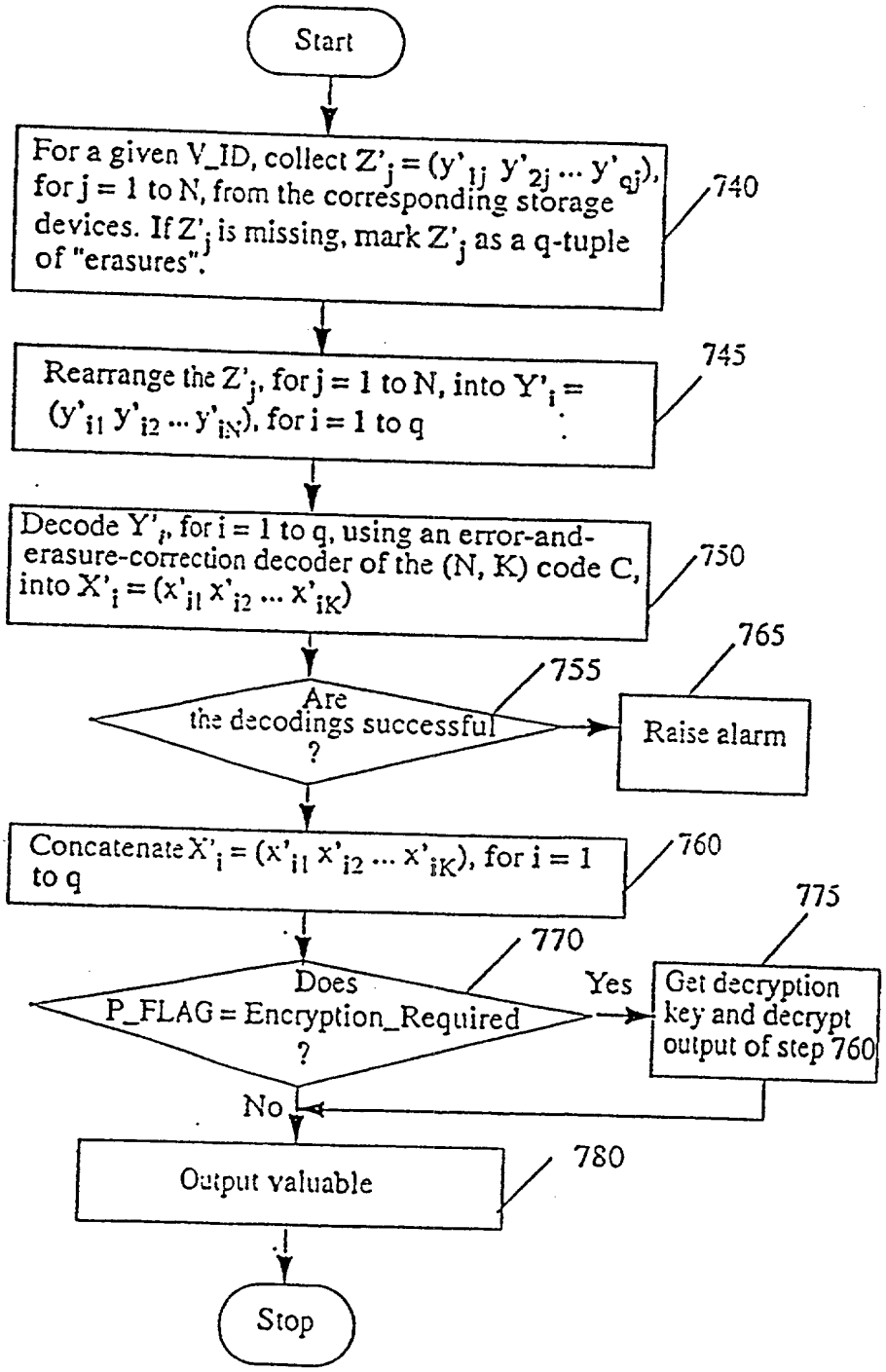
Start
For a given V_ID, collect $Z'_j = (y'_{1j}\ y'_{2j} \cdots y'_{qj})$, for j = 1 to N, from the corresponding storage devices. If Z'_j is missing, mark Z'_j as a q-tuple of "erasures".
740
Rearrange the Z'_j, for j = 1 to N, into $Y'_i = (y'_{i1}\ y'_{i2} \cdots y'_{iN})$, for i = 1 to q
745
Decode Y'_i, for i = 1 to q, using an error-and-erasure-correction decoder of the (N, K) code C, into $X'_i = (x'_{i1}\ x'_{i2} \cdots x'_{iK})$
750
Are the decodings successful ?
755
Raise alarm
765
Concatenate $X'_i = (x'_{i1}\ x'_{i2} \cdots x'_{iK})$, for i = 1 to q
760
Does P_FLAG = Encryption_Required ?
770
Yes
Get decryption key and decrypt output of step 760
775
No
Output valuable
780
Stop

FIG. 8

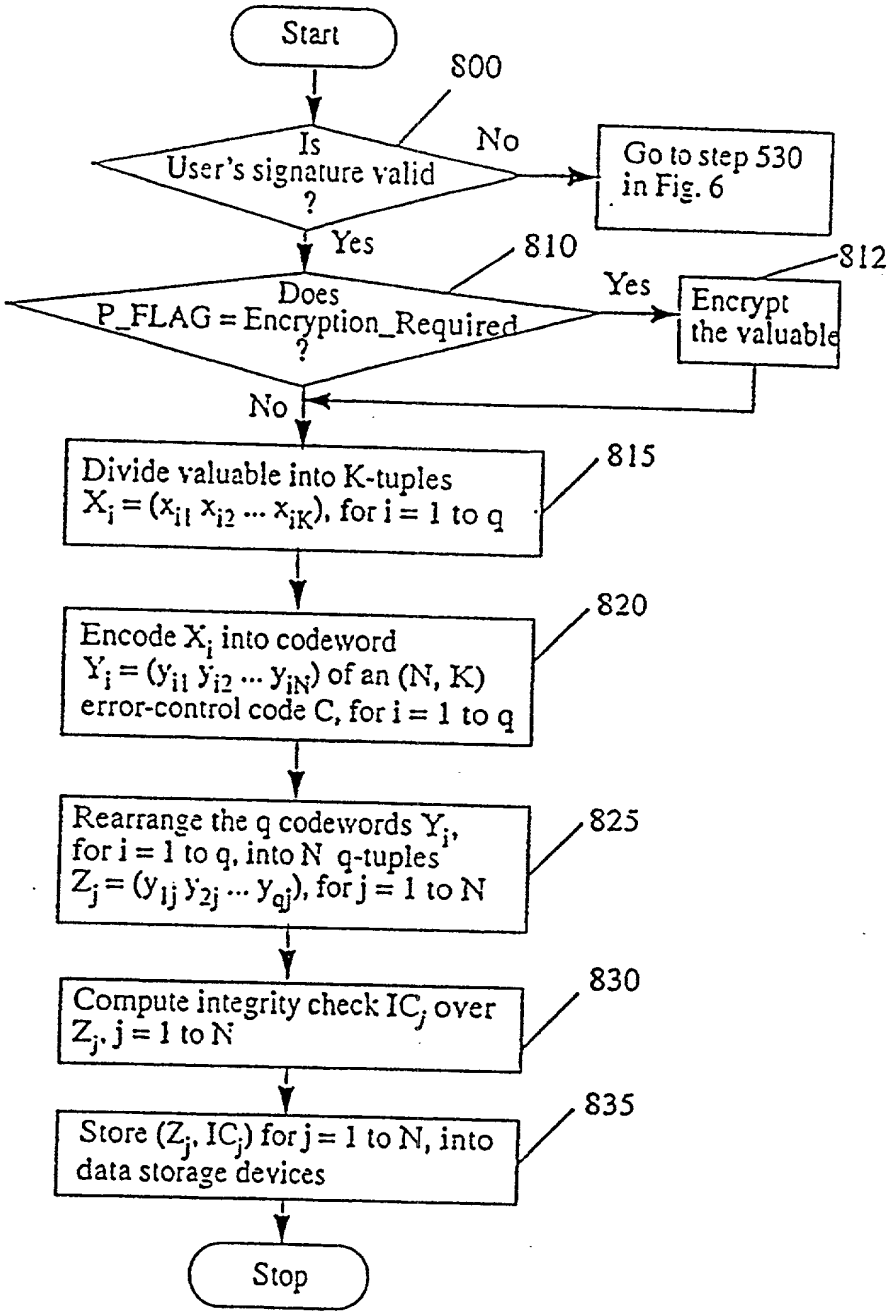
Start
800
Is User's signature valid ?
No
Go to step 530 in Fig. 6
Yes
810
Does P_FLAG = Encryption_Required ?
Yes
812
Encrypt the valuable
No
Divide valuable into K-tuples $X_i = (x_{i1}\ x_{i2} \cdots x_{iK})$, for i = 1 to q
815
Encode X_i into codeword $Y_i = (y_{i1}\ y_{i2} \cdots y_{iN})$ of an (N, K) error-control code C, for i = 1 to q
820
Rearrange the q codewords Y_i, for i = 1 to q, into N q-tuples $Z_j = (y_{1j}\ y_{2j} \cdots y_{qj})$, for j = 1 to N
825
Compute integrity check IC_j over Z_j, j = 1 to N
830
Store (Z_j, IC_j) for j = 1 to N, into data storage devices
835
Stop

FIG. 9

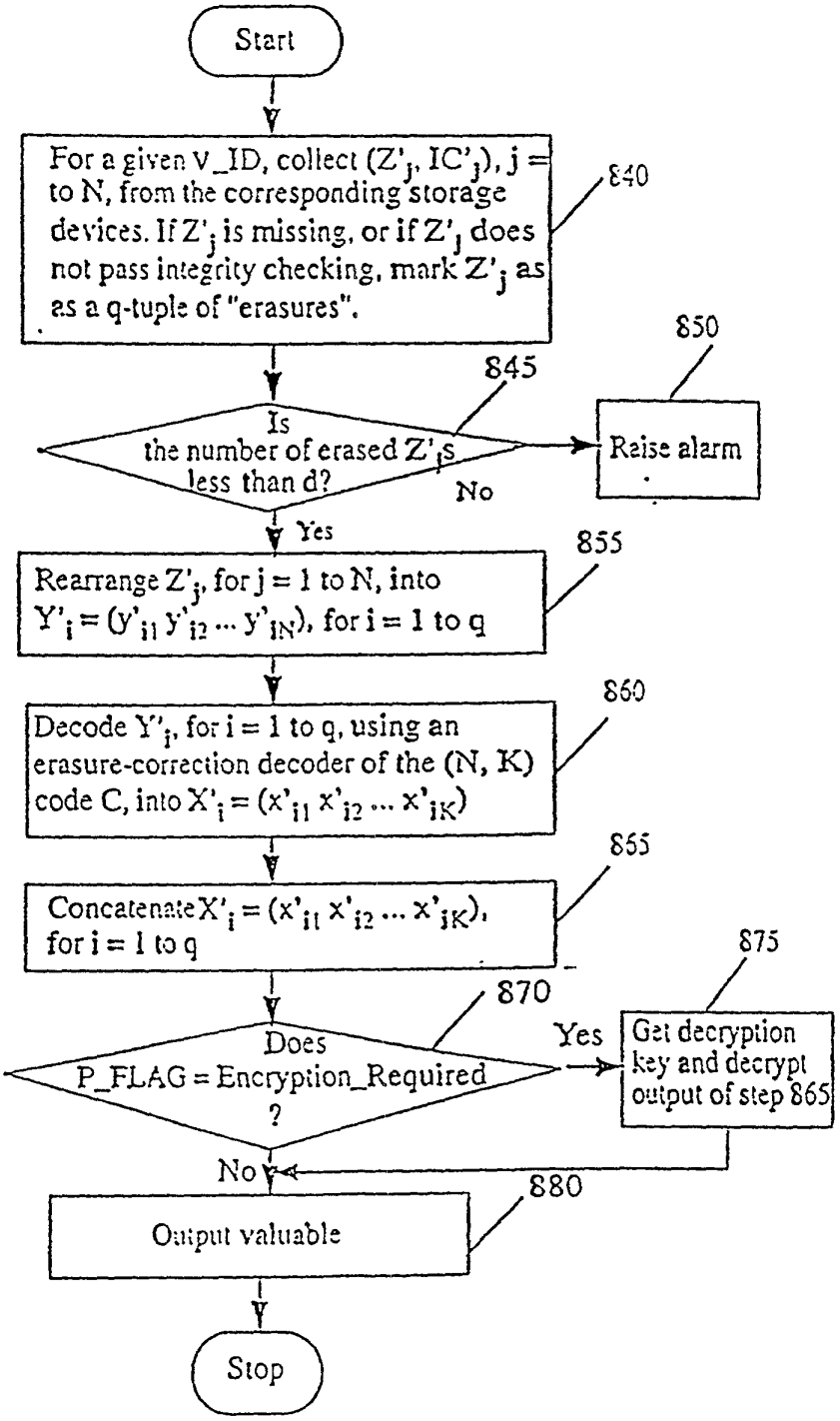
Start
For a given V_ID, collect (Z'j, IC'j), j = to N, from the corresponding storage devices. If Z'j is missing, or if Z'j does not pass integrity checking, mark Z'j as as a q-tuple of "erasures".
840
Is the number of erased Z'j s less than d?
845
No
Raise alarm
850
Yes
Rearrange Z'j, for j = 1 to N, into Y'i = (y'i1 y'i2 ... y'iN), for i = 1 to q
855
Decode Y'i, for i = 1 to q, using an erasure-correction decoder of the (N, K) code C, into X'i = (x'i1 x'i2 ... x'iK)
860
Concatenate X'i = (x'i1 x'i2 ... x'iK), for i = 1 to q
865
Does P_FLAG = Encryption_Required ?
870
Yes
Get decryption key and decrypt output of step 865
875
No
Output valuable
880
Stop

FIG. 10

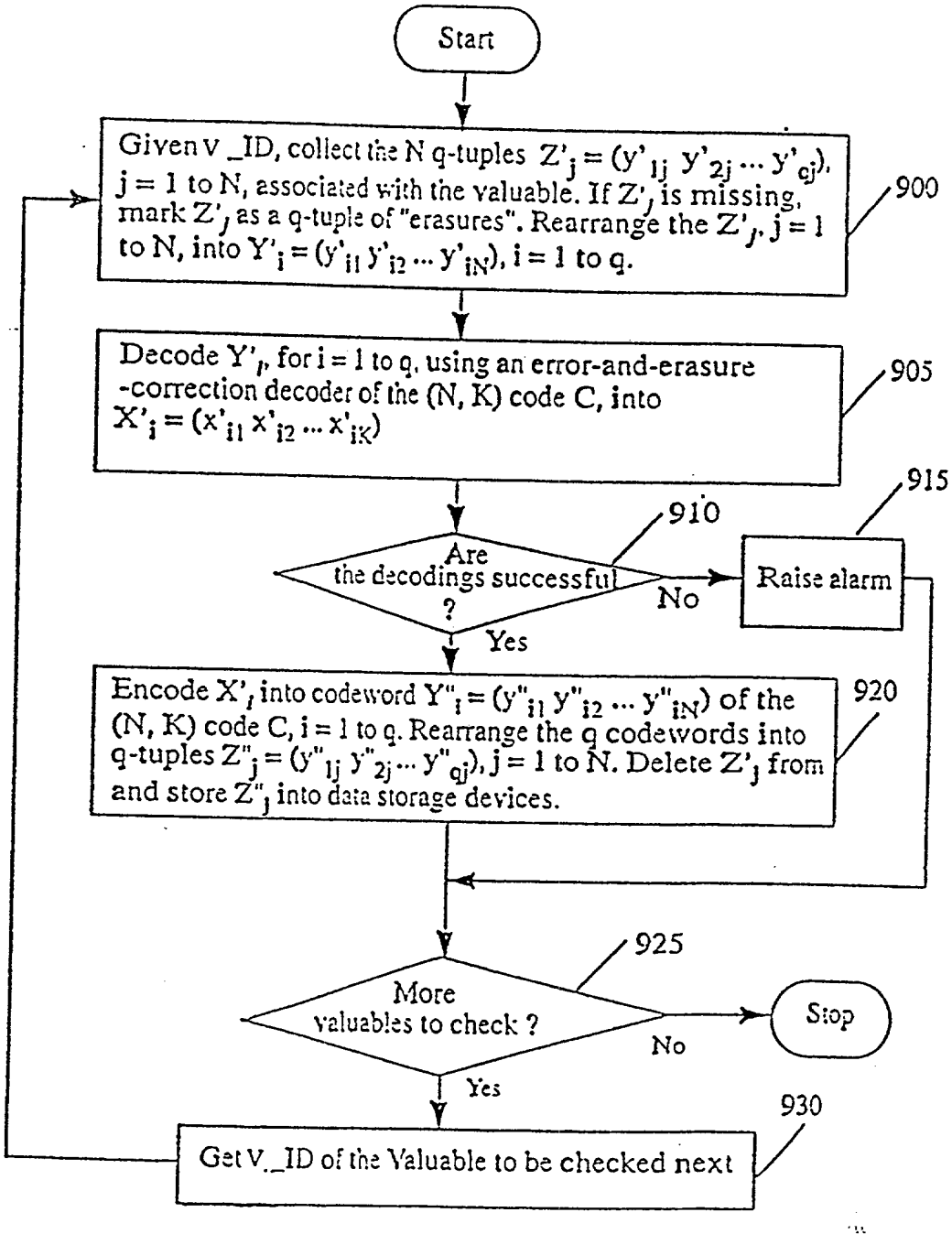
Start
Given v_ID, collect the N q-tuples $Z'_j = (y'_{1j}\ y'_{2j} \cdots y'_{qj})$, j = 1 to N, associated with the valuable. If Z'_j is missing, mark Z'_j as a q-tuple of "erasures". Rearrange the Z'_j, j = 1 to N, into $Y'_i = (y'_{i1}\ y'_{i2} \cdots y'_{iN})$, i = 1 to q.
900
Decode Y'_i, for i = 1 to q, using an error-and-erasure -correction decoder of the (N, K) code C, into $X'_i = (x'_{i1}\ x'_{i2} \cdots x'_{iK})$
905
Are the decodings successful ?
910
No
Raise alarm
915
Yes
Encode X'_i into codeword $Y''_i = (y''_{i1}\ y''_{i2} \cdots y''_{iN})$ of the (N, K) code C, i = 1 to q. Rearrange the q codewords into q-tuples $Z''_j = (y''_{1j}\ y''_{2j} \cdots y''_{qj})$, j = 1 to N. Delete Z'_j from and store Z''_j into data storage devices.
920
More valuables to check ?
925
No
Stop
Yes
Get v_ID of the Valuable to be checked next
930

FIG. 11

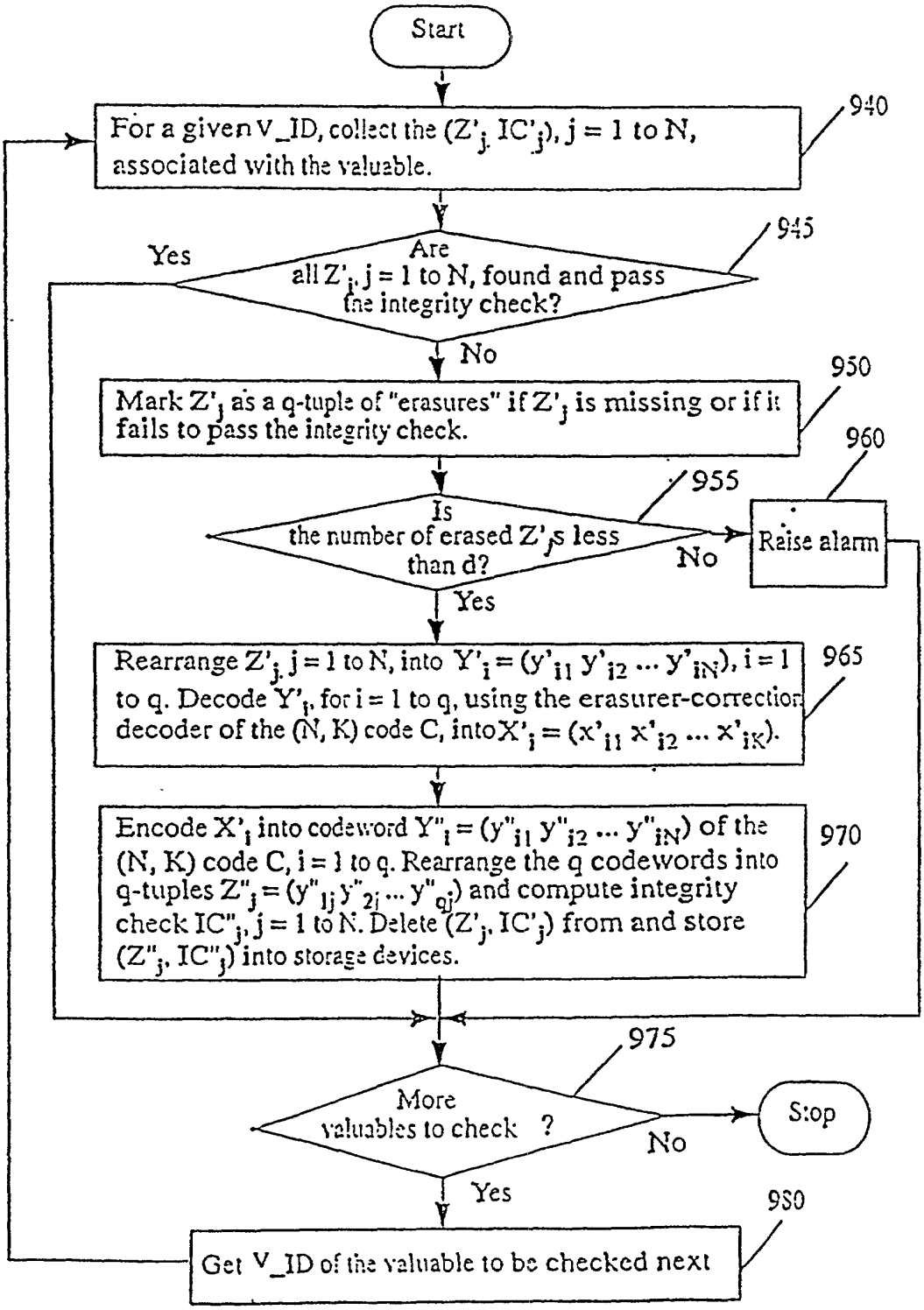
Start
For a given V_ID, collect the (Z'j, IC'j), j = 1 to N, associated with the valuable.
940
Are all Z'j, j = 1 to N, found and pass the integrity check?
945
Yes
No
Mark Z'j as a q-tuple of "erasures" if Z'j is missing or if it fails to pass the integrity check.
950
960
Is the number of erased Z'j s less than d?
955
No
Raise alarm
Yes
Rearrange Z'j, j = 1 to N, into Y'i = (y'i1 y'i2 ... y'iN), i = 1 to q. Decode Y'i, for i = 1 to q, using the erasurer-correction decoder of the (N, K) code C, into X'i = (x'i1 x'i2 ... x'iK).
965
Encode X'i into codeword Y"i = (y"i1 y"i2 ... y"iN) of the (N, K) code C, i = 1 to q. Rearrange the q codewords into q-tuples Z"j = (y"1j y"2j ... y"qj) and compute integrity check IC"j, j = 1 to N. Delete (Z'j, IC'j) from and store (Z"j, IC"j) into storage devices.
970
975
More valuables to check ?
No
Stop
Yes
980
Get V_ID of the valuable to be checked next

FIG. 12

Attorney Docket No.

MERCHANT & GOULD P.C.

United States Patent Application

COMBINED DECLARATION AND POWER OF ATTORNEY

As a below named inventor I hereby declare that: my residence, post office address and citizenship are as stated below next to m name; that

I verily believe I am the original, first and sole inventor (if only one name is listed below) or a joint inventor (if plural inventors are named below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: ___

The specification of which
a. ☐ is attached hereto
b. ☑ was filed on as application serial no. and was amended on (if applicable) (in the case of a PCT-filed application) described and claimed in international no. PCT/SG98/00003 filed 16 JAN 1998 and as amended on (if any), which I have reviewed and for which I solicit a United States patent.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code Federal Regulations, § 1.56 (attached hereto).

I hereby claim foreign priority benefits under Title 35, United States Code, § 119/365 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date befor that of the application on the basis of which priority is claimed:

a. ☑ no such applications have been filed.
b. ☐ such applications have been filed as follows:

FOREIGN APPLICATION(S), IF ANY, CLAIMING PRIORITY UNDER 35 USC § 119			
COUNTRY	APPLICATION NUMBER	DATE OF FILING (day, month, year)	DATE OF ISSUE (day, month, year)
ALL FOREIGN APPLICATION(S), IF ANY, FILED BEFORE THE PRIORITY APPLICATION(S)			
COUNTRY	APPLICATION NUMBER	DATE OF FILING (day, month, year)	DATE OF ISSUE (day, month, year)

I hereby claim the benefit under Title 35, United States Code, § 120/365 of any United States and PCT international application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in th manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, § 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application.

U.S. APPLICATION NUMBER	DATE OF FILING (day, month, year)	STATUS (patented, pending, abandoned)

I hereby claim the benefit under Title 35, United States Code § 119(e) of any United States provisional application(s) listed below:

U.S. PROVISIONAL APPLICATION NUMBER	DATE OF FILING (Day, Month, Year)

(03)

I hereby appoint the following attorney(s) and/or patent agent(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith:

Albrecht, John W.	Reg. No. 40,481	Lacy, Paul E.	Reg. No. 38,946
Anderson, Gregg I.	Reg. No. 28,828	Larson, James A.	Reg. No. 40,443
Batzli, Brian H.	Reg. No. 32,960	Liepa, Mara E.	Reg. No. 40,066
Beard, John L.	Reg. No. 27,612	Lindquist, Timothy A.	Reg. No. 40,701
Berns, John M.	Reg. No. 43,496	Lycke, Lawrence E.	Reg. No. 38,540
Black, Bruce E.	Reg. No. 41,622	McAuley, Steven A.	Reg. No. 46,084
Branch, John W.	Reg. No. 41,633	McDonald, Daniel W.	Reg. No. 32,044
Bremer, Dennis C.	Reg. No. 40,528	McIntyre, Jr., William F.	Reg. No. P-44,921
Bruess, Steven C.	Reg. No. 34,130	Mueller, Douglas P.	Reg. No. 30,300
Byrne, Linda M.	Reg. No. 32,404	Pauly, Daniel M.	Reg. No. 40,123
Carlson, Alan G.	Reg. No. 25,959	Phillips, John B.	Reg. No. 37,206
Caspers, Philip P.	Reg. No. 33,227	Plunkett, Theodore	Reg. No. 37,209
Chiapetta, James R.	Reg. No. 39,634	Prendergast, Paul	Reg. No. 46,068
Clifford, John A.	Reg. No. 30,247	Pytel, Melissa J.	Reg. No. 41,512
Cochran, William W.	Reg. No. 26,652	Qualey, Terry	Reg. No. 25,148
Daignault, Ronald A.	Reg. No. 25,968	Reich, John C.	Reg. No. 37,703
Daley, Dennis R.	Reg. No. 34,994	Reiland, Earl D.	Reg. No. 25,767
Dalglish, Leslie E.	Reg. No. 40,579	Schmaltz, David G.	Reg. No. 39,828
Daulton, Julie R.	Reg. No. 36,414	Schuman, Mark D.	Reg. No. 31,197
DeVries Smith, Katherine M.	Reg. No. 42,157	Schumann, Michael D.	Reg. No. 30,422
DiPietro, Mark J.	Reg. No. 28,707	Scull, Timothy B.	Reg. No. 42,137
Edell, Robert T.	Reg. No. 20,187	Sebald, Gregory A.	Reg. No. 33,280
Epp Ryan, Sandra	Reg. No. 39,667	Skoog, Mark T.	Reg. No. 40,178
Glance, Robert J.	Reg. No. 40,620	Spellman, Steven J.	Reg. No. 45,124
Goggin, Matthew J.	Reg. No. 44,125	Stoll-DeBell, Kirstin L.	Reg. No. 43,164
Golla, Charles E.	Reg. No. 26,896	Sumner, John P.	Reg. No. 29,114
Gorman, Alan G.	Reg. No. 38,472	Swenson, Erik G.	Reg. No. 45,147
Gould, John D.	Reg. No. 18,223	Tellekson, David K.	Reg. No. 32,314
Gregson, Richard	Reg. No. 41,804	Trembath, Jon R.	Reg. No. 38,344
Gresens, John J.	Reg. No. 33,112	Underhill, Albert L.	Reg. No. 27,403
Hamre, Curtis B.	Reg. No. 29,165	Vandenburgh, J. Derek	Reg. No. 32,179
Hillson, Randall A.	Reg. No. 31,838	Wahl, John R.	Reg. No. 33,044
Holzer, Jr., Richard J.	Reg. No. 42,668	Weaver, Karrie G.	Reg. No. 43,245
Johnston, Scott W.	Reg. No. 39,721	Welter, Paul A.	Reg. No. 20,890
Kadievitch, Natalie D.	Reg. No. 34,196	Whipps, Brian	Reg. No. 43,261
Karjcker, Shaukat	Reg. No. 34,049	Wickhem, J. Scot	Reg. No. 41,376
Kastelic, Joseph M.	Reg. No. 37,160	Williams, Douglas J.	Reg. No. 27,054
Kettelberger, Denise	Reg. No. 33,924	Witt, Jonelle	Reg. No. 41,980
Keys, Jeramie J.	Reg. No. 42,724	Wu, Tong	Reg. No. 43,361
Knearl, Homer L.	Reg. No. 21,197	Xu, Min S.	Reg. No. 39,536
Kowalchyk, Alan W.	Reg. No. 31,535	Zeuli, Anthony R.	Reg. No. 45,255
Kowalchyk, Katherine M.	Reg. No. 36,848		

I hereby authorize them to act and rely on instructions from and communicate directly with the person/assignee/attorney/firm/ organizatio who/which first sends/sent this case to them and by whom/which I hereby declare that I have consented after full disclosure to be represented unless/until I instruct Merchant & Gould P.C. to the contrary.

Please direct all correspondence in this case to Merchant & Gould P.C. at the address indicated below:

Merchant & Gould P.C.
P.O. Box 2903
Minneapolis, MN 55402-0903

*2355
2*

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statemen may jeopardize the validity of the application or any patent issued thereon.

2	Full Name Of Inventor	Family Name: HU	First Given Name: Jian	Second Given Name:
0	Residence & Citizenship	City: Singapore	State or Foreign Country: Singapore SGX	Country of Citizenship: China
1	Post Office Address	Post Office Address: Blk 302 #11-531, Clementi Ave. 4,	City: Singapore	State & Zip Code/Country: Singapore 120302
Signature of Inventor 201:			Date: 3 July 2000	

2	Full Name Of Inventor	Family Name: BAO	First Given Name: Feng	Second Given Name:
0	Residence & Citizenship	City: Singapore	State or Foreign Country: Singapore SGX	Country of Citizenship: China
2	Post Office Address	Post Office Address: 37 West Coast Park #04-06	City: Singapore	State & Zip Code/Country: Singapore 127653
Signature of Inventor 202:			Date: 10/07/00	

2	Full Name Of Inventor	Family Name: DENG	First Given Name: Huije	Second Given Name:
0	Residence & Citizenship	City: Singapore	State or Foreign Country: Singapore SGX	Country of Citizenship: Singapore
3	Post Office Address	Post Office Address: 2 Namly Rise	City: Singapore	State & Zip Code/Country: Singapore 267110
Signature of Inventor 203:			Date: 3 July 2000	

§ 1.56 Duty to disclose information material to patentability.

(a) A patent by its very nature is affected with a public interest. The public interest is best served, and the most effective patent examination occurs when, at the time an application is being examined, the Office is aware of and evaluates the teachings of all information material to patentability. Each individual associated with the filing and prosecution of a patent application has a duty of cand and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section. The duty to disclose information exists with respect to each pending claim until the claim is canceled or withdrawn from consideration, or the application becomes abandoned. Information material to the patentability of a claim that is canceled or withdrawn from consideration need not be submitted if the information is not material to the patentability of any claim remaining under consideration in the application. There is no duty to submit information which is not material to the patentability o any existing claim. The duty to disclose all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office in the manner prescribed by §§ 1.97(b)-(d) and 1.98. However, no patent will be granted on an application in connection with which fraud on the Office was practiced or attempted or the duty of disclosure was violated through bad faith or intentional misconduct. The Office encourages applicants to carefully examine:

(1) prior art cited in search reports of a foreign patent office in a counterpart application, and

(2) the closest information over which individuals associated with the filing or prosecution of a patent application believe any pending claim patentably defines, to make sure that any material information contained therein is disclosed to the Office.

(b) Under this section, information is material to patentability when it is not cumulative to information already of record or being made of record in the application, and

(1) It establishes, by itself or in combination with other information, a prima facie case of unpatentability of a clai

or

(2) It refutes, or is inconsistent with, a position the applicant takes in:

(i) Opposing an argument of unpatentability relied on by the Office, or

(ii) Asserting an argument of patentability.

A prima facie case of unpatentability is established when the information compels a conclusion that a claim is unpatentable under the preponderance of evidence, burden-of-proof standard, giving each term in the claim its broadest reasonable construction consistent with th specification, and before any consideration is given to evidence which may be submitted in an attempt to establish a contrary conclusion o patentability.

(c) Individuals associated with the filing or prosecution of a patent application within the meaning of this section are:

(1) Each inventor named in the application:

(2) Each attorney or agent who prepares or prosecutes the application: and

(3) Every other person who is substantively involved in the preparation or prosecution of the application and who i associated with the inventor, with the assignee or with anyone to whom there is an obligation to assign the application.

(d) Individuals other than the attorney, agent or inventor may comply with this section by disclosing information to the attorney, agent, or inventor.